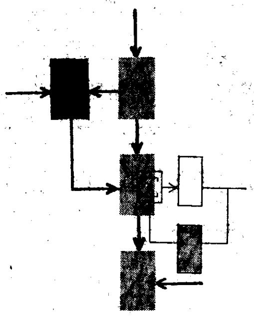


TECHNICAL MEMORANDUM

PAPERS ON AUTOMATIC PROGRAMMING FOR NUMERICALLY CONTROLLED MACHINE TOOLS

Douglas T. Ross

6873-TM-3

January 7, 1958

Servomechanisms Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE 39, MASSACHUSETTS

Department of Electrical Engineering

85

TECHNICAL MEMORANDUM

PAPERS ON AUTOMATIC PROGRAMMING FOR NUMERICALLY CONTROLLED MACHINE TOOLS

Douglas T. Ross
6873-TM-3
January 7, 1958

This document reports the results of work made possible through the support extended the Massachusetts Institute of Technology, Servomechanisms Laboratory, by the United States Air Force, Air Materiel Command, under Contract No. AF33(038)-24007, M.I.T. Project No. D.I.C. 6873. It is published for technical information only and does not represent recommendations or conclusions of the sponsoring agency. When U. S. Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related U. S. Government procurement operation, the U. S. Government thereby incurs no responsibility or obligation whatsoever; and the fact that the U. S. Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation or conveying any right or permission to manufacture, use, or sell any patented invention that may be in any way related thereto.

Approved by: Donald F. Clements
Donald F. Clements

Servomechanisms Laboratory
Department of Electrical Engineering
Massachusetts Institute of Technology

ABSTRACT

This technical memorandum is a reproduction of two papers written by Mr. Douglas T. Ross, Head of the Computer Applications Group of the Servomechanisms Laboratory at M.I.T.

Presentation of the first paper was made at the Third Annual Contour Machining Conference, October 23, 24, and 25, 1957, at the Ambassador Hotel, Los Angeles, California, and the second paper at the Association for Computing Machinery Session of the Indianapolis meeting of the American Association for the Advancement of Science on December 28, 1957.

The first paper is descriptive, covering the major developments in programming for numerical control up to the present day. The "APT System" is introduced for combining the human designer, the general purpose computer, and the numerically controlled machine tool for production of complex parts.

The second paper is primarily tutorial, describing the philosophy behind the current research developments. The "systematized solution" concept, which is the distinguishing feature of the recent work, appears to be applicable to problem solving in other fields as well as numerical control.

Together these papers present a condensed summary of the present activities in this field at M.I.T., and provide a glimpse of some future possibilities.

TABLE OF CONTENTS

SECTION I

SOME RECENT DEVELOPMENTS IN AUTOMATIC PROGRAMMING FOR NUMERICALLY CONTROLLED MACHINE TOOLS

SECTION II

DEVELOPMENT OF A RESEARCH EFFORT IN THE AUTOMATIC PROGRAMMING OF NUMERICALLY CONTROLLED MACHINE TOOLS

LIST OF FIGURES

SECTION I

SECTION II

SECTION I

SOME RECENT DEVELOPMENTS IN AUTOMATIC PROGRAMMING FOR NUMERICALLY CONTROLLED MACHINE TOOLS*

A. INTRODUCTION

In the past several years the importance of numerical control as a new production technique has been widely publicized. By controlling a physical process in response to numbers, numerical control provides a link between the growing field of data processing and the production of actual parts. It is now possible to use the artificial, mathematical records and computations of the design process for the creation of pilot or production models of complex parts. In order to achieve some of the benefits of this new manufacturing technique, it is necessary to create a means for passing directly from the plans for a part in the mind of the designer to the detailed, numerical instructions which will cause the desired part to be produced. This is the goal of automatic programming, a new aspect of the technology which has recently come into being.

This paper describes some recent developments in automatic programming of numerically controlled machine tools, using large, general purpose digital computers. In addition to describing some of the major aspects of these systems, a few of the problems which govern their design are described. Although it will be some time before the potentialities of this new technology can be fully realized, members of the aircraft industry are now supporting a joint programming effort to achieve a limited automatic programming system in the near future for contour machining. The techniques are also applicable to many other production processes.

---

*A paper to be presented at the Third Annual Contour Machining Conference, October 23, 24, 25, 1957, Ambassador Hotel, Los Angeles.

B. HISTORY

A new era in small-lot production of machined parts opened in 1952 with the completion of the first fully automatic numerically controlled machine tool by the Servomechanisms Laboratory of M.I.T. The control system for this machine tool (shown in Fig. 1) represented the culmination of several years of intensive research, privately sponsored at first, and then supported by the Air Materiel Command of the U. S. Air Force. Following the completion of the machine tool system itself, a series of studies were made to analyze the problem of preparation of data for numerically controlled machine tools, and to compare the economics of this new method of production with standard machining practices. One part of this study was the development of a set of subroutines for the Whirlwind I computer at M.I.T. to assist in the preparation of numerical data in the form required by the machine tool control system.

Contoured parts can be made on the numerically controlled machine by approximating curved segments by sequences of closely spaced straight lines in space. In order to meet tolerance specifications, extremely large numbers of segments are required for even the simplest curves. In addition to being tedious, calculation of these many points by hand using desk calculators is not only expensive and time consuming, but also unreliable since many mistakes in the rather complicated processing can be made. For this reason it was recognized early that since many of the computations are repetitious, the computing abilities of large scale, general purpose digital computers would be ideally suited to the problem of data preparation for numerical control.

The library of subroutines, the development of which was itself a non-trivial problem, shouldered a major portion of the routine calculating load, but left a number of things to be desired. First of all, the nature of the subroutines which were written, combined with the wide variety of parts which were to be made by numerical control, meant that the library soon became extremely large and the selection of the appropriate routine to perform a given function was itself a difficult task. It was not unusual to find that a routine needed to be modified slightly to fit a particular application once it was found. In addition it was necessary that the part programmer know not only tooling and machining problems, but be a computer programmer as well. Therefore, although the subroutine library did achieve some considerable success in reducing the amount of routine labor in preparing data for numerical control, this technique still was not viewed as a desirable method for long term use.

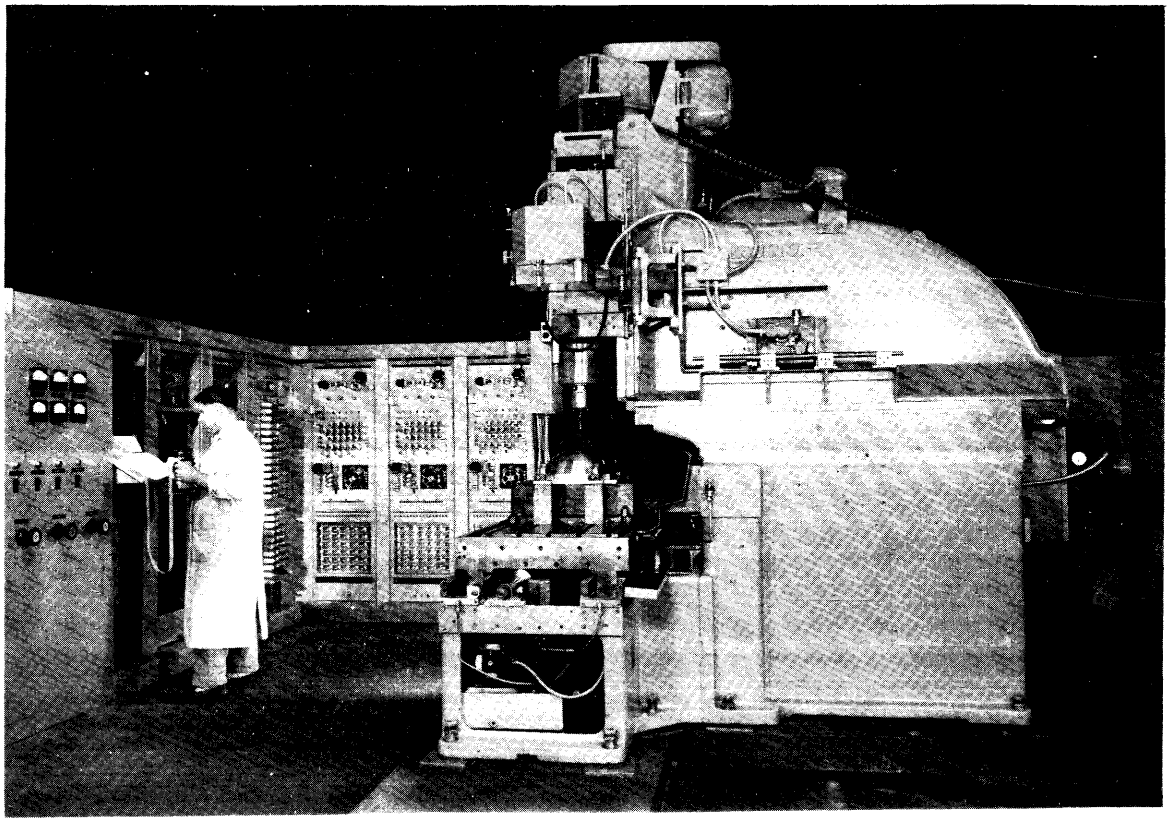
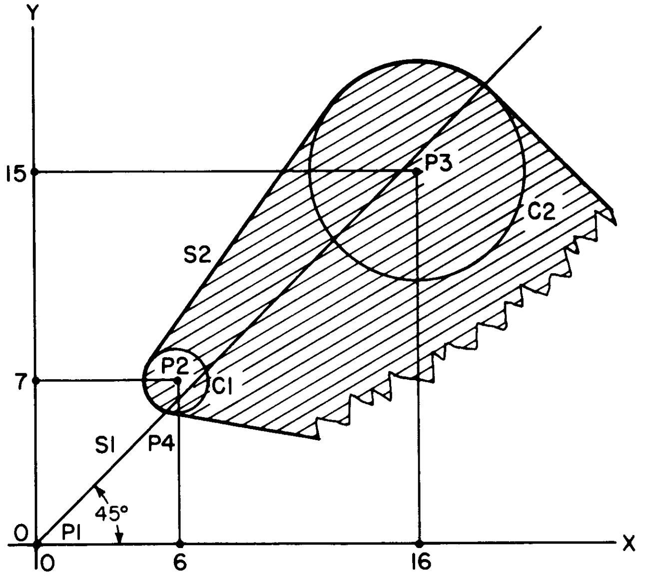


Fig. 1 The M.I.T. Numerical Controlled Machine Tool

C. INTRODUCTION OF AUTOMATIC PROGRAMMING

About the time the shortcomings of the subroutine library approach to data processing were recognized, notable successes were being achieved in the field of automatic programming of general purpose digital computers. Automatic programming entails the generation of detailed, specific instructions in the language of the computer, from statements made in a specially designed, easy to use, less specific language. In 1955 the basic features of automatic programming were applied to the problem of data processing for numerical control in the form of a pilot program written by Arnold Siegel of the Digital Computer Laboratory of M.I.T. The program operated on the Whirlwind I computer and automatically produced punched paper control tapes for the M.I.T. Numerically Controlled Milling Machine for arbitrary 2-dimensional parts which could be specified by means of straight lines and circles. The language, although cryptic, was highly mnemonic with "s" standing for straight line, "c" for circle, etc., and special words such as TOOL RAD. and FEEDRATE for specifying the various parameters of the machine tool control problem. An example of the use of this language for the programming of a sample part is shown in Fig. 2.

A major advantage of automatic programming systems is that they can be made to assist the part programmer in stating his problem as well as carrying out



```
P1 = 0,0
P2 = 6,7
C1 = P2|2
P3 = 15,16
C2 = P3|5
S1 = P1|45°
P4 = NS1|C1
S2 = TC1|TC2
P5 = FS1|C2
..............
..............
 0,P1
 15,P4
 15,C1,S2
 15,S2,C2
 15,C2,P5
...............
```

Fig. 2 Sample Part Program, Siegel Language the steps to solve the problem once it is stated. Thus in the example, some of the straight lines and circles are constructed implicitly by statements in the language itself and are not explicitly stated as inputs to the problem. This ability to define new curves on the basis of points and curves already specified in the language statement is one of the most powerful aspects of automatic programming systems, since it frees the part programmer from the labor (and inherent errors) of solving for portions of the part which are known only implicitly by their relationship to other portions of the part. Using an automatic programming system of this type, it may be possible to produce a part only on the basis of the designer's original specifications, without any intermediate human calculations.

The economic studies of the application of numerical control to actual production parts had shown clearly that although the new technique was virtually competitive with standard techniques, a disproportionate amount of time and expense was consumed in the data processing part of the problem. The initial success of the pilot program indicated that further development of automatic programming could make the possibilities of numerical control not only practically realizable, but more than competitive with standard techniques. Research into the problems of automatic programming of 3-dimensional parts, to be produced on 3 and 5 axis numerically controlled machine tools is now under way at the Servomechanisms Laboratory, M.I.T. This project has been active for approximately one year, and the results to date are the subject of the remainder of this paper.

D. THE APT SYSTEM CONCEPT

The first problem which confronted the new project was to determine the scope and direction of the research effort. An examination of the 2-dimensional pilot program disclosed very quickly that although the program had met its limited objectives admirably, the techniques which had been used in solving the 2-dimensional problem were not applicable to 3 dimensions, and the automatic programming features could not readily be extended to handle complicated parts. It was also recognized that many levels of sophistication were possible for automatic programming systems of this type, and that it would be best to progress gradually through several stages. Each stage would be characterized by the computer system taking over a larger portion of the detailed work of preparing data for numerical control, so that the burden on the part programmer would be progressively lessened.

The result of these deliberations was an overall concept of the application of automatic programming techniques to numerically controlled machine tools. The name "APT SYSTEM, " an abbreviation for Automatically Programmed Tool system, was coined to embody the all important, governing principles. The research program must recognize that the disparate characteristics of the human programmer, the general purpose computer, and the numerically controlled machine tool cannot be considered one at a time, but must be linked together in an overall system concept. It was also felt that the most practical way to limit the objectives of the effort without stifling its vitality was to establish a hierarchy of successive APT systems, each one being characterized by more and more sophistication in the human input language.

Figure 3 shows the hierarchy of APT systems as presently envisaged. Each level translates the instructions of the part programmer, stated in successively more convenient languages, and carries out these instructions by automatic programming of lower systems in the hierarchy. The lowest level of the hierarchy has a minimum of automaticity and utilizes the general purpose computer as a calculating aid through the use of a library of subroutines. The next level, the APT II system, allows the part programmer to specify and produce a part by statements in APT language which are made in terms of space curves, rather than individual points as in the APT I system. The highest level of APT system which has so far been determined concretely is the APT III system, in which the statements in the language are made in terms of regions of the part, i.e., entire portions of machined surfaces. In all of these systems it is the function of the general purpose computer which embodies the APT system to process the statements in the convenient APT language into detailed numerical statements in the language of the machine tool director, which will cause the tool to make the required thousands of short, straight line motions. Another important feature of all APT systems is an auxiliary output from the computer which is in the same APT language which the numan has used to program the part, or in more appropriate (apt!) language forms, so that the human can monitor the performance of the automatic system and correct his programming mistakes without the expenditure of valuable shop time and materials. One such form of output for the human is to have the computer draw 3-dimensional pictures of the part being made on an output oscilloscope for viewing by the human. Figure 4 shows a number of views of parts drawn by the prototype Whirlwind I computer programs which have been written to demonstrate APT II and APT III systems. The pictures show the many straight-line motions the tool would make to produce each part.

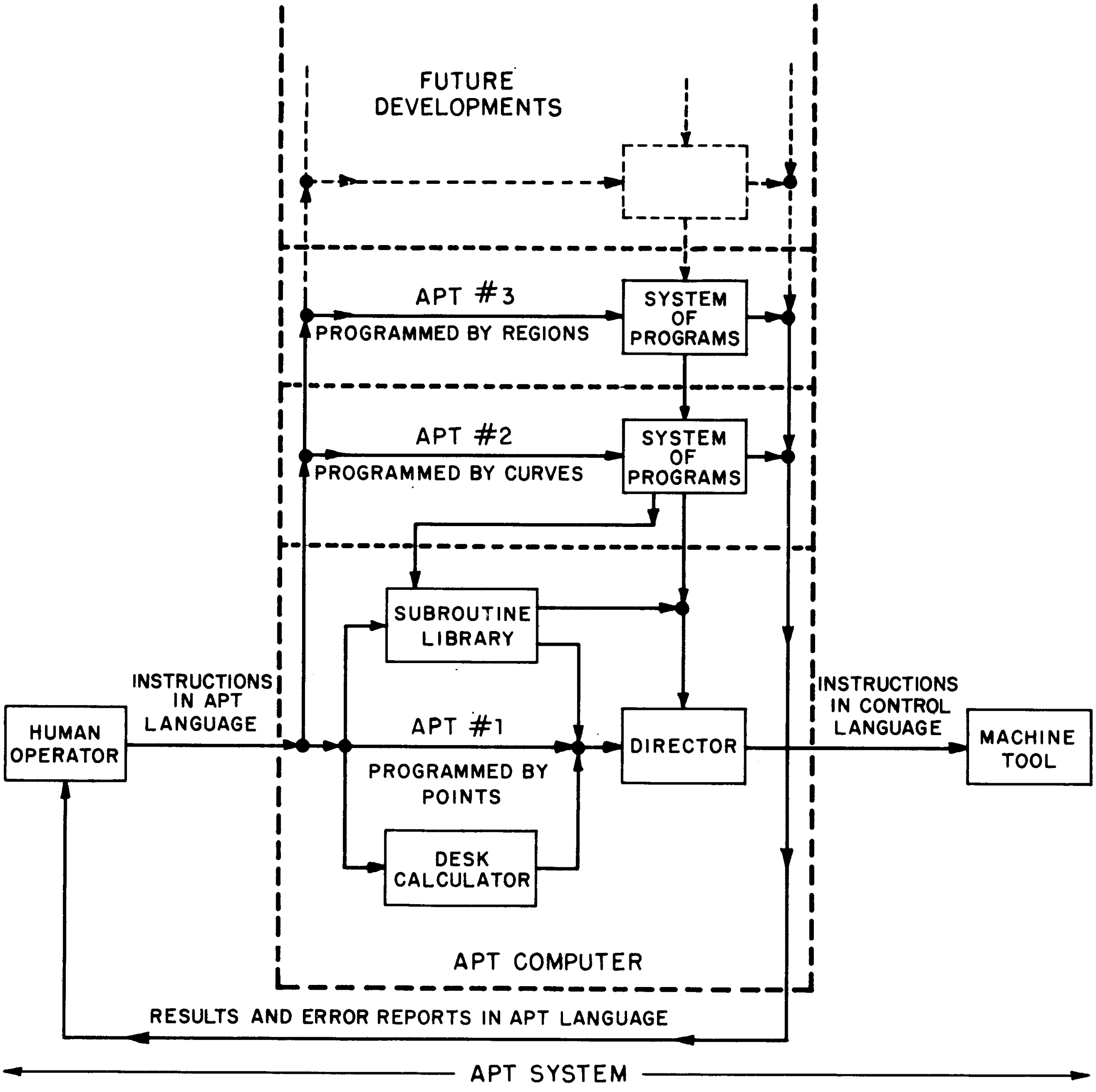


Fig. 3 The APT System Concept

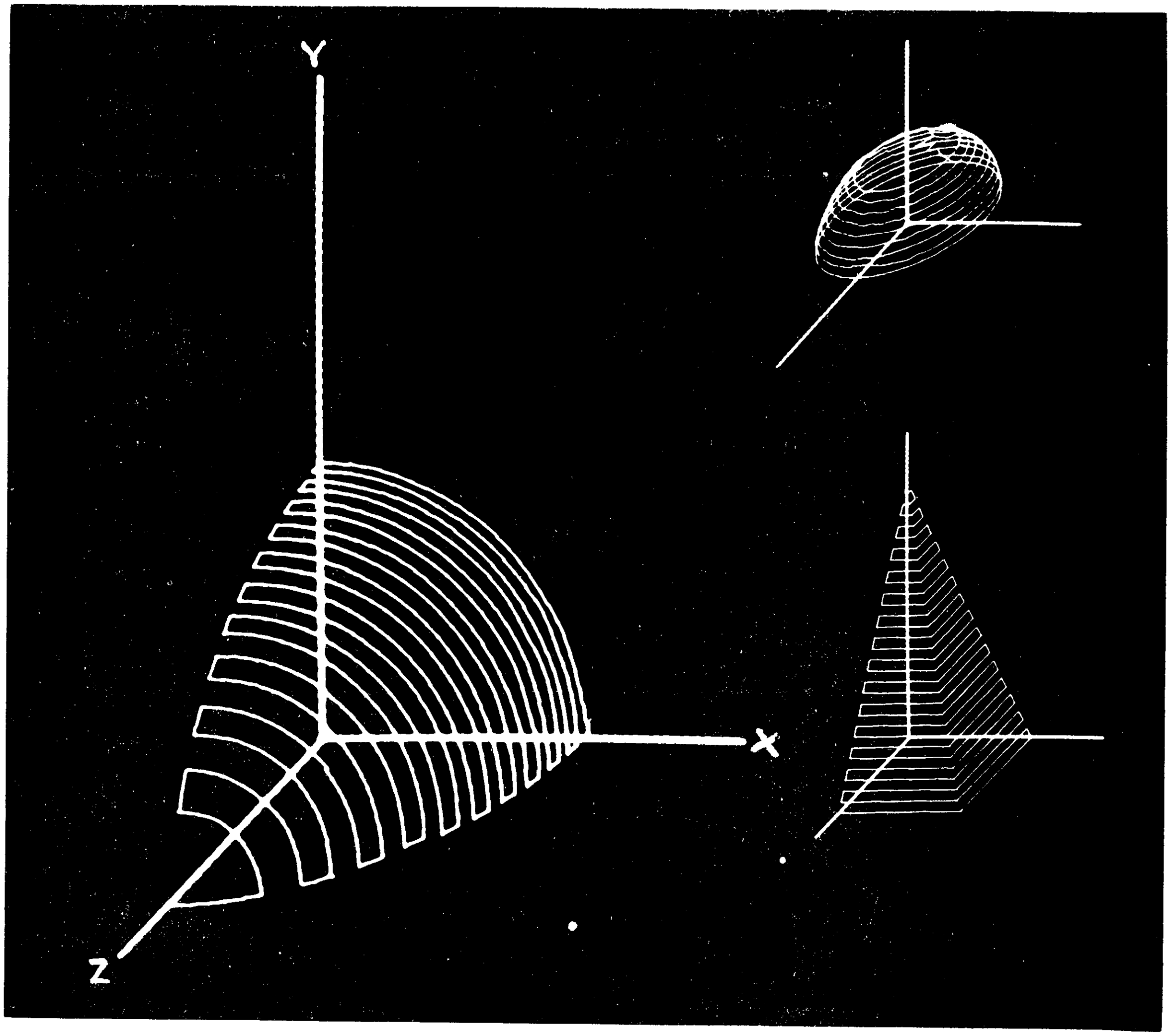


Fig. 4 Pictures Drawn by the Whirlwind Computer APT II and APT III Programs

E. RESEARCH OBJECTIVES

The external features of APT systems have now been briefly described, but in order to engender confidence that this highly sophisticated type of data processing system can indeed be achieved on a practical scale, it is necessary to look behind the scenes briefly at the general motivations and conditions under which these systems are being developed. The primary objective is that APT systems of any level must be capable of continued growth. The use of automatic programming for producing parts is an entirely new technology, and much remains to be learned about the many intricate aspects of this new technique. It will never be possible to say that a particular APT program is finished because the very use of that system will introduce new methods and new types of parts which lie just outside the scope of the system and which must be incorporated into the system by modification. Recognizing this as a fact of life for all future APT systems, it is of paramount importance that they be designed from the very beginning in such a way that their character and content can be dynamic and in a state of continual change without adversely affecting either their usefulness or requiring unreasonable amounts of effort to incorporate new features. It is extremely difficult to ensure orderly progress for the continued development of complicated systems which cannot even be envisaged in their entirety, but a number of features can be built into the systems which show real promise toward satisfying these needs.

The first step in building in growth possibilities comes almost automatically, but is of great importance. It is possible to program a general purpose computer to simulate all of the elements of a specially designed computer. By using these programming techniques, a general purpose computer can be made to act like an entirely different computer, one which is designed specifically to solve a certain type of problem. This technique is admirably suited to the development of APT systems since it is possible to design the special purpose computer to have a specified APT language as input and machine tool director or human picture language as output, and special techniques can be developed for storing intermediate surface, tooling, and metal-cutting information in a suitably designed "memory." The concept of simulating a special purpose computer provides the basic structural framework within which the various portions of the APT system can be developed. The resulting set of programs also inherently has the required characteristic of being programmable, a feature which is of primary importance because the APT system is to solve whatever problem the part programmer specifies in his APT language statement. Just as the large scale general purpose digital computers of today consist of many separate packages which are scattered around the room and interconnected by cables, the simulated computer consists of many individual programs with their own particular functions to perform interconnected by logical pathways. Thus, it is possible to allow the APT system to grow by replacing and altering individual boxes in the system without affecting other boxes, and with limited expenditures of effort.

Before such a simulated computer can be established, however, it is necessary to have an analysis of the many problems inherent in the complicated data processing prodedure for numerical control, which will enable the problem to be broken into separate portions, and which will itself allow for orderly growth of the system. For lack of better words, the term "systematized solution" has been coined to represent this idea of a solution to a problem area which can be particularized to solve any individual problem which falls within that area. Systematized solutions have already been developed as a part of the APT system research effort for the solution of the geometrical problems involved in the APT II and APT III methods of programming by curves and programming by regions. These methods contain the essence of the problem of moving a cutting tool through space to produce a specified curve or region, and are independent of the particular surfaces and dimensions involved. To solve a particular problem, it is necessary only to supply the particular surface information in a routine fashion, and then the complicated solution is achieved automatically. These systematized solutions supply the necessary methodology to be built into the calculating portions of the simulated computer.

Similarly systematized solutions are now under development to allow the orderly growth of the language and language translation portions of the APT systems. With the completion of these systematized solutions it will be possible to add new ways of describing parts and programming tool motions to the system conveniently and without disrupting the other portions of the system. It seems probable that as experience grows with the use of these systems, the APT language will prove to be the most changeable portion of the system so that a systematized solution to the problem of incorporating language translation changes is of great importance.

F. ACTIVITIES WITH THE AIA SUBCOMMITTEE FOR NUMERICAL CONTROL

The Subcommittee for Numerical Control of the Aircraft Industries Association has been extremely active for some time in studying the problems of the introduction of numerical control into aircraft manufacturing facilities. With the imminent delivery to various aircraft plants of a large number of numerically controlled machine tools, purchased by the Air Materiel Command, this group is studying ways to achieve compatibility among the various plants, and methods for processing the large amounts of data which the new equipment requires. Several of the large aircraft companies have developed numerical control programming systems using general purpose digital computers in order to get the new machine tools into action at the earliest possible moment.

The preliminary results of the M.I.T. research studies were presented to the Subcommittee in early 1957, and it was decided that the many aims of the Subcommittee for industry-wide compatability of data processing could be met by joining in a cooperative programming effort. The various plants have pooled their computer programming manpower and resources to develop a preliminary automatic programming system based on the principles of the prototype M.I.T. systems. Figure 5 is a diagram of the 2D-APT II system (2-dimensional input language, programmed by space curves) showing the various tasks which are now being programmed for the IBM 704 computer. Over 20 programmers from 16 aircraft plants are presently expending most of their effort in this work. It was estimated that a preliminary version of the system could be operating in October, but the unpredictable problems of blending the 2D-APT II work in with the necessary operations of the individual plants resulted in uneven starting dates for the various tasks, so that it now appears that January is a more realistic target date.

The language of the initial system is designed to be primarily 2-dimensional, although the calculating portions of the system are capable of 3-dimensional work. The preliminary language has been chosen primarily for ease of translation and does not yet represent the best thinking of the group. The language is, however, quite mnemonic and symbolic, and in Fig. 6 shows a sample program as it would appear on the part programmer's manuscript, for the initial system.

G. FUTURE POSSIBILITIES

The possibilities of automatic programming are limited only by imagination and economics. Although the development of automatic programming systems is indeed expensive, the potentialities have hardly been scratched and it is impossible to predict accurately the net cost of automatic programming once it is put to extensive use. It is a well-known fact, however, that research and design which go into production parts are extremely expensive items, so that any automatic system which can take over even a portion of these functions has a good chance of paying

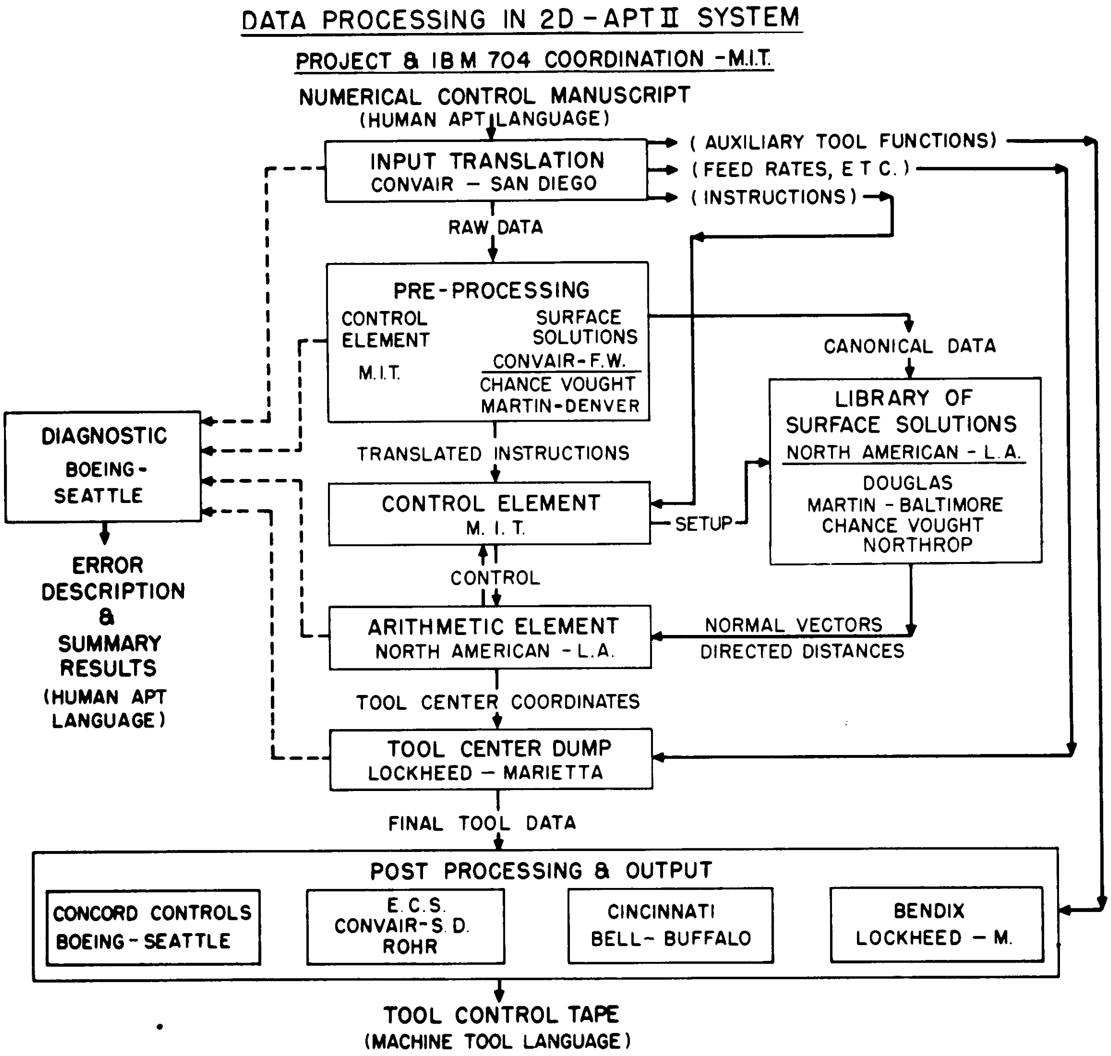
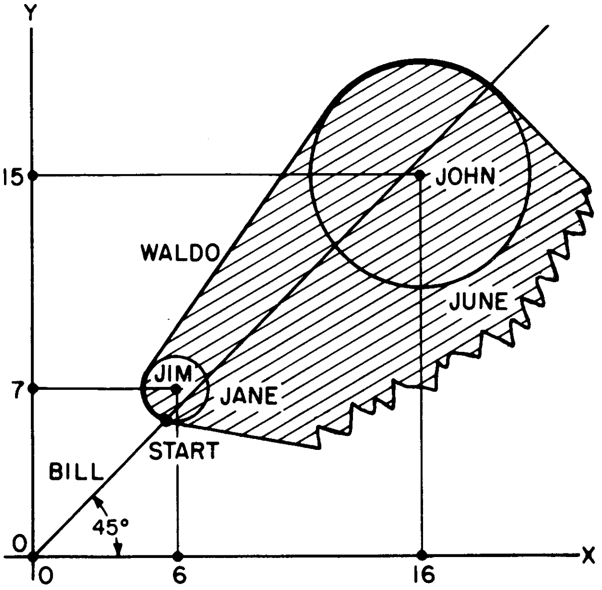


Fig. 5 The 2D-APT-II Joint Programming Effort



```
JIM   = POINT/+6., +7. $$
JANE  = CIRCL/JIM, +2. $$
JOHN  = POINT/+16.,+15. $$
JUNE  = CIRCL/JOHN, +5. $$
BILL  = LINE/+0.,+0.,+45.$$
START = SMAL Y, INT OF/BILL,JANE $$
WALDO = SMAL X, LARG Y, TAN TO/JANE,JUNE $$
        ..................
        ....................
    FROM /+0., +0. $$
    FED RT = +15. $$
    GO TO / START $$
    GO LEFT/JANE,WALDO $$
    GO FWD / WALDO, JUNE $$
NEAR, GO CLW /JUNE, BILL $$
        ......................
```

Fig. 6 Sample Part Program, 2D-APT-II Language

its own way.

Any attempt to put into an automatic system some of the design functions now performed by highly skilled humans is immediately confronted with a number of very difficult problems. The mechanical and computational steps through which a design progresses are a fundamental problem, but existing techniques of numerical analysis and related topics, if diligently applied, can probably yield ready solutions. A much more difficult problem, one which is unique to automatic programming systems, is that of determining the parameters and decisions from which a designer begins his design, and finding a way for him to express these quantities in a form which can be used to instruct the automatic programming system. It is not possible to build real intelligence into an automatic system so that it is necessary that the designer pass on to the system sufficient information for the automatic system to determine the necessary parameters. This is the problem of language design, the problem of determining a vocabulary and rules for making statements to the automatic system which will convey the design problem in an unsolved but complete form, from the mind of the human to the computer system. Since this type of problem never arose before the advent of automatic programming systems, very little is known at the present time about this area and much research remains to be done

Assuming steady progress in the development of automatic programming systems, it is not improbable that within the next decade or so it will be possible to obtain a sample part from a system having specified only a few structural dimensions and the various forces, stresses, and vibrations to which the part will be subjected with all of the intermediate steps of design and drafting, model making and testing essentially eliminated. The problem of fairing between various surfaces will surely be one of the first road-blocks to fall and a number of related problem areas are currently being worked on in various places in industry. It is interesting to note that aerodynamics, hydrodynamics, and stress analysis are all subject to similar mathematical laws, the laws of streamlines, so that a breakthrough in any one of these areas could affect the other areas.

The members of industry should be patient when it comes to automatic programming. A common reaction is that the remarkable possibilities which automatic programming offers are so attractive that it is momentarily forgotten that this is still an infant technology and initial systems will necessarily have limited capabilities. It is desirable that industry support this type of research through their own industrial organizations, so that the practical problems of production and design are covered adequately. Although the APT techniques which are currently being developed are for 3-dimensional contour milling, the concepts and ideas are applicable wherever numerical control is used. It is hoped that the presentation of these ideas will stimulate thinking and action in other parts of industry.

SECTION II

DEVELOPMENT OF A RESEARCH EFFORT IN THE AUTOMATIC PROGRAMMING OF NUMERICALLY CONTROLLED MACHINE TOOLS*

A. INTRODUCTION AND ABSTRACT

A promising new technology has been introduced by the development of machines which can perform complex tasks in response to coded instructions. The most familiar machines of this type are the modern large-scale digital computers, but the concept of controlling processes by means of numbers can be applied to many non-mathematical machines as well. The most sophisticated production tools of this type are the numerically controlled machine tools which permit continuous control of the motion and orientation of a cutting tool in three dimensions. Highly complex three-dimensional parts can be sculptured automatically on these machines by "programming" the required motions by means of appropriate sequences of numbers recorded on punched paper tape. Since humans are not used to thinking of machined parts in terms of the thousands of numbers required, numerical control cannot be fully exploited until the tedious and repetitious tasks of data preparation can automatically be accomplished.

Current research at M.I.T. is aimed at programming digital computers to "speak" the machine tool's language and "understand" a specialized form of written English, so that machined parts can be made automatically merely by "telling" the computer-tool system what to do. The collection of computer programs which accomplishes this task is called an APT System (an abbreviation for Automatically Programmed Tool System), and prototype systems have been operating at M.I.T.

---

* A paper presented at the Association For Computing Machinery Session of the Indianapolis Meeting of the American Association for the Advancement of Science, December 28, 1957. (To be submitted to Journal of the Association for Computing Machinery.)

for several months. The APT system approach shows such promise that a number of aircraft companies have pooled manpower and resources, under the technical coordination of the M.I.T. project, to produce such systems for industry-wide use, even though many already have made heavy investments in computer-tool systems of their own. The first system, called 2D-APT II Phase I, is scheduled for completion in early 1958 and will represent 10 to 15 man years of cooperative work.

New techniques are used in the development of APT systems which represent a significant advance in automatic programming system design. The heart of the new approach is the "systematized solution" concept which allows any of a class of related problems to be solved by merely fleshing out a single universal skeleton program. The systematized solution concept is a powerful problem-solving technique which can fruitfully be applied in many areas of research. In the case of APT system design, diligent application of the new principles greatly clarifies the obscure problems which arise in spatial geometry, language design, language translation, and system organization. It is not improbable that continued work along these lines can lead to a "design machine" which will assist in the design process itself, and then automatically produce a part to meet the specified requirements.

Although the immediate importance of these developments seems to be technological, the true impact may well lie more in the scientific world. The advent of the programmed machine, for the first time permits abstract knowledge to be applied directly to physical processes, so that it is expected that fundamental new insights into the nature of the design and problem-solving processes will evolve from these studies.

B. HISTORY

The first numerically controlled machine tool was introduced by the Servomechanisms Laboratory, M.I.T., in 1952, and since that time many similar machines have been designed and are now being put into actual production use. The machine tool portion of the M.I.T. system was an ordinary 3-dimensional milling machine modified slightly to provide the necessary connections for feedback control circuitry. The other major portion of the system is called the numerical control director and consists of electronic equipment for reading coded instructions from punched paper tape and actuating the control mechanisms so that the cutting tool follows a prescribed path. The input information on the punched paper tape consists of blocks of four numbers, the first three give the incremental motion which is to be made in each of the three axes, and the fourth provides a measure of the time between successive points and thus controls the feedrate of the cutter.

Arbitrary 3-dimensional surfaces may be sculptured on the numerically controlled milling machine by approximating curved portions by sufficiently many tool center locations such that the deviation between the generated surface and the desired surface is less than a specified tolerance. Because the input information to the system is so elementary, the types and complexities of surfaces which can be machined is virtually unlimited. In order to produce a complex part, however, many thousands of incremental cuts must be specified so that a large data processing problem results.

C. NUMERICAL CONTROL PROGRAMMING

The process of writing down the sequence of numerical instructions which will cause the numerically controlled machine tool to produce a specified part is called programming for numerical control, or more simply part programming. There are a number of distinct operations which must be performed in transforming the information provided by the designer of the part into the detailed numerical instructions which are required, and a number of separate disciplines are involved. Geometry must be used to locate specified points, curves, and surfaces, and other mathematical operations are necessary to determine the sequence of tool center locations which will meet the required tolerance when connected by straight lines. Questions of metallurgy also arise in the choice of cutting tool, spindle speed, and feedrates which are employed in making cuts. There are also problems of fixture design and tooling, since metal removal is a violent process and the work must properly be held and supported if tolerances are to be met in the finished part. When working to close tolerances it is even possible for the stresses induced in the metal in the process of cutting to cause sufficient warpage after holding clamps are removed to invalidate the part. All of these problems must be considered in programming for numerical control since once the instructions are specified, the entire cutting operation will proceed automatically.

Because of the many different aspects of the numerical control programming problem, many of the decisions which must be made in part programming are of great importance to the finished product and require all of the judgement and skill which the part programmer can bring to bear. These important decisions, however, lie almost entirely in the metallurgical aspects of the problem and the mathematical and geometric considerations are for the most part tediously routine.

Since it is not natural for human beings to think of machined parts in terms of the thousands of numbers required by numerical control, they are not only slow but also unreliable in carrying out accurately the many detailed steps required to program directly in the machine's numerical input language.

Although it will be some time before any considerable portion of the important metallurgical decisions can be made automatically, a vigorous research effort is now underway at the Servomechanisms Laboratory to develop techniques for performing all of the detailed mathematical and geometric operations of numerical control programming automatically. A new type of automatic programming system is being developed in which general purpose digital computers are programmed to "speak" the machine tool's language and "understand" a specialized form of written English so that machined parts can be made automatically merely by "telling" the computer-tool system what to do. The collection of computer programs which accomplishes this task is called an APT system and prototype systems have been operating at M.I.T. for several months. This paper describes the kind of research effort which has been employed in APT system development.

The techniques used in APT systems are subtly but fundamentally different from those used in most present-day automatic programming systems. In order to demonstrate clearly the unique features of this new type of system, it is necessary first to review those techniques which have been applied to automatic programming problems (and the numerical problem in particular) in the past.

D. SUBROUTINE LIBRARIES

Data preparation for numerical control is a complicated, detailed and time-consuming task, and if human beings had to perform all of the various operations themselves it is doubtful that numerical control could be considered as an important production method. Those very aspects of the numerical control programming problem which are unnatural and difficult for the human to perform reliably are precisely the type which are highly appropriate for modern general-purpose digital computers. The standard operations of determining cut lengths so that a specified tolerance can be met, and of calculating the necessary corrections for tool center locations, etc., are mathematical operations which, although they are not necessarily simple, are nonetheless routine and require no judgement on the part of the part programmer. Transforming numbers into the coded form required by the numerically controlled machine tool is also difficult for a human to perform, but very well suited to the capabilities of the computer. It is therefore possible to set up a collection of computer programs, called a subroutine library, which may be pieced together to solve almost any routine problem which arises in numerical control programming automatically and at high speed, and with a great increase of reliability. Such a system of computer programs can greatly reduce the work load of the part programmer, although his full faculties are still required for making decisions which depend upon metallurgy and sequencing. In order to utilize the subroutine library in this form, however, the part programmer must be familiar with computer programming as well as part programming, since the only way that the various subroutines can be united to apply to a particular problem is by means of computer coding.

E. AUTOMATIC PROGRAMMING

The burden on the part programmer may be further eased by employing still another kind of modern powerful computer programming technique. It is possible to write programs which cause general purpose computers to perform purely executive functions under the control of an easy-to-use input language. Thus, the mathematical and data handling subroutines of the library may be augmented by non-mathematical routines which allow the part programmer to use the computer and yet know nothing about computer coding. If a special language is designed with words specially chosen to match the problems of a particular programming area such as part design, and if a set of computer programs is written for transforming statements made in that language into basic machine language, then the collection of computer programs are said to form an automatic programming system. Automatic programming, then, consists of expressing a problem to an intermediate system of computer programs in an easy-to-use language, after which the further programming required to produce a program in basic machine language is performed automatically by the system. Several systems of this type, in which the computer automatically sets up and interconnects subroutines selected from the library, have been devised for numerical control programming.

If a limited variety of problems are to be solved (i.e. if a limited variety of machined parts is to be made), then the subroutine library with automatic programming is a highly efficient system, and can be made to satisfy completely the requirements of the part programmer. The part programmer need only learn the natural, specially designed language and then all of the mathematical and geometrical details of numerical control programming will automatically be taken care of.

If, however, as is usually the case, an important aspect of the use of numerical control is that more and more elaborate machined parts can be made, then it is necessary for the subroutine library and the corresponding automatic programming executive routines to be in a continual state of flux and development. As new classes of machined parts are contemplated, additions must be made to the library and to the executive routines to allow the specification of these more complicated surfaces within the framework of the original library system. As the library grows in size and complexity, it becomes increasingly difficult to devise efficient executive routines for selecting and properly interconnecting the appropriate subroutines. As the library grows it also becomes increasingly apparent that many of the subroutines in the library are meant to perform essentially the same function with only minor differences. For example, there might be separate subroutines in the library for calculating incremental tool motion over a sphere, over a cylinder, over a cone, over an airfoil section, etc. Therefore as the tasks which are to be performed by the system become more comprehensive and complex, the subroutine library with its many types of subroutine and many versions of each type becomes cumbersome and inefficient.

F. THE SYSTEMATIZED SOLUTION

The primary difficulty in using a library of subroutines with automatic programming, the fact that there are so many pieces so nearly alike, provides the key to an improved approach. If there is a large class of subroutines, all of which perform essentially the same function but merely with respect to different surfaces or conditions, would not a superior solution result if the common function of these subroutines could be separated out and a single generalized program written which could be fleshed out to fit any particular case? In solving any individual problem, there are many different guide-posts which direct the researcher to a satisfactory solution for that single problem, whereas when an entire class of problems is considered a much more intensive investigation is required since only the "essence" of the problem is left. Therefore, although this concept does indeed appear to have real merit, the solution to the common function, a "systematized solution" which can be made to solve any of the separate problems, is usually quite difficult to find. Although more penetrating analyses are required to solve classes of problems than single problems, the concept of a systematized solution is very encompassing and shows real promise in overcoming the difficulties which are apparent in the straightforward library of subroutines concept. Marked successes have already been registered in the solution of the geometric problems associated with numerical control programming.

As part of the current M.I.T. research project, successful systematized solutions have been developed for automatically determining the sequence of incremental motions required to move a cutting tool over an arbitrary space curve or an arbitrary surface in space in such a way that the resulting machined curve or surface will be within tolerance over its entire extent. A detailed description of these systematized solutions would be much too technical for this paper, but some of the important aspects of the research that went into their development can be outlined. An example is given in the appendix which illustrates in a simple way the complete development of a systematized solution for a related problem.

In seeking a systematized solution, i.e., in solving the "essence" of a class of problems, the seemingly important characteristics which give each problem its individuality must be purposefully ignored and the essential basic features common to all of the problems must be uncovered. It is often necessary that the techniques used must be impersonal and of a mechanistic nature since only the common structure of the problems can be used. Although the common structure represents the relationships and interconnections between the basic elements which compose the problem, it may frequently be advantageous to base the mechanistic solution upon intermediate calculable quantities or properties instead of upon the basic elements themselves. All of these general principles were employed in developing the systematized solutions for tool motion for use in APT systems.

G. NORMAL VECTOR AND DIRECTED DISTANCE

Although it is possible to conceive of basing motion of a cutting tool over a surface directly upon the mathematical equation which defines the surface, a much more general solution was found to be possible by introducing two fundamental geometric properties for which subroutines could easily be written. Instead of working directly with the equations for surfaces, two subroutines called the normal vector subroutine and the directed distance subroutine are written for each surface type (see Fig. 1). A normal vector program for a surface has as inputs the coordinates of a point on the surface and supplies as output a unit vector perpendicular to the surface at that point. A directed distance program for a surface has as inputs the coordinates of a point in space and the components of a vector specifying a direction in which the surface is to be viewed from that point. The directed distance pro-

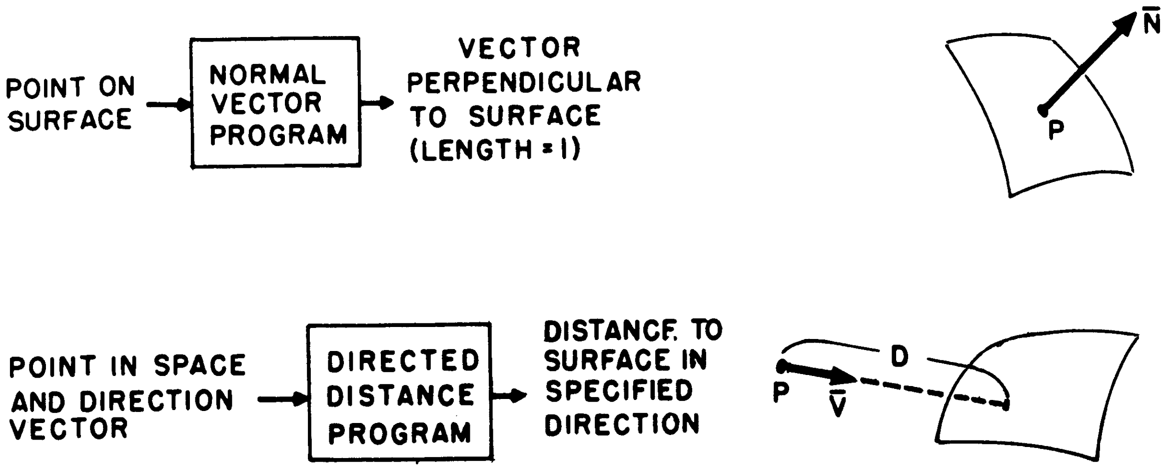


Fig. 1 Normal Vector and Directed Distance Programs

gram provides as output the distance to the surface from the specified point looking in the specified direction.

It is usually most efficient to write special normal vector and directed distance subroutines for each type of surface, but it is also possible to devise general-purpose iterative normal vector and directed distance programs so that subroutines for functional definition of the surface itself is all that is required. Therefore, as long as a surface can be defined by means of a program, normal vectors and directed distances can be computed even though special analyses for them may not be known.

It has been found that these special descriptive properties of surfaces supply all of the information needed for constructing mechanistic, systematized solutions to the geometric problems which arise in numerical control programming. The normal vector program provides information about the orientation of the surface, and since the tangent plane to the surface at that point is determined by the set of all vectors perpendicular to the normal vector, it is possible to "move" on the surface approximately by moving along a selected tangent vector. The directed distance program is very useful since if the distance to the surface is known in a specified direction, it is possible to "move" directly to the surface. The combination of normal vector and directed distance programs, therefore, allow approximate motions to be corrected exactly, so that motion over arbitrary surfaces is possible without any build-up of errors. Normal vector programs are also useful for calculating curvatures of surfaces, etc., and directed distance programs can be used in many ways to control step-by-step iterative calculations with respect to surfaces, since it is possible to "look ahead" and determine whether a given condition has been met.

Using the fundamental geometric properties of normal vectors and directed distances, single general purpose programs have been written which form the skeleton for a program which will solve the tool motion problem for any particular surface. It is only necessary to flesh out the skeleton by selecting the appropriate normal vector and directed distance subroutines and inserting them into the appropriate locations in the general skeleton program. Thus the concept of a systematized solution has greatly simplified the geometric part of the numerical control programming problem. Notice that the subroutine library has not been completely eliminated, (normal vector and directed distance subroutines are still required for each type of surface), but it certainly has been reduced in complexity. The chaotic lack of structure of the large subroutine library has been replaced by the serenity of a homogeneous, smaller library with fixed skeleton programs in which all of the difficult problems have been solved once and for all. An ordered and controlled growth of the APT system capabilities can be envisioned since it is easy to add new surface types to the simple library.

H. SIMULATION OF SPECIAL APT COMPUTERS

The next question to be asked is: How does the systematized solution concept affect the design of the executive routines which are necessary to apply automatic programming features?

The important fact that a systematized solution can be "programmed" merely by fleshing out a fixed skeleton program which is written once and for all, may be combined with a powerful programming technique called interpretive programming to result in an automatic programming system which possesses a substantial advantage over the ordinary subroutine library approach. Through the use of interpretive programming it is possible to write programs which "interpret" coded instructions and therefore can be programmed themselves, i.e. these programs are programmable. Systematized solutions are ideally suited to interpretive programming since they are also programmable in the sense that to solve any particular problem it is necessary merely to insert the appropriate subroutines. If systematized solutions are available, it is possible to design and simulate special purpose computers which can perform complex tasks as their basic programmable functions. Just as a general-purpose computer can perform the basic arithmetic operations, add, subtract, multiply, and divide, when instructed which numbers to use, a simulated computer can calculate intersections, motions, and other pertinent quantities when instructed which curves or surfaces to use.

The systematized solutions for motion over arbitrary curves and surfaces have been programmed in such a way that they simulate the function of the arithmetic element of a simulated APT computer. Other interpretive programs act like a control element and automatically set up the appropriate subroutines for surfaces (which correspond to the numbers of an ordinary computer) as the calculations progress. The APT computer is not simulated by means of a single interpretive program which represents one computer with all the desired characteristics. In actuality, the overall APT computer consists of a collection of special purpose computers each of which is designed to do its own particular kind of job most efficiently. Each simulated computer is based upon a systematized solution which captures the essence of its own particular type of problem and solves it with greatest efficiency. Some of these computers work independently and operate directly upon the APT language statements made by the human programmer. Others, however, control and are controlled by other computers by means of special intermediate automatic programming. Thus, for instance, the program which calculates tool motion along space curves may be used to provide a "first pass" around the boundary of a space region, and the motion over the remainder of the region may then be automatically generated by another Arithmetic Element Program which calculates motion over a surface.

An important feature of this subdivision of the APT computer's function is that it is possible to achieve maximum efficiency at all levels of the highly complicated data processing without sacrificing convenience in programmed expressions. Continued growth of the overall APT system is made feasible since more comprehensive problems are solved by building on the previous research work, and it is not usually necessary to discard programs and start from scratch again since each individual computer already represents the best solution to its own class of problems. In order to achieve this type of permanence for the programs as they are developed, however, it is necessary that the initial analysis be very penetrating and soundly based upon properties and concepts which are truly fundamental to the class of problems being solved. If a program does become obsolete, the cause can usually be traced to an imperfect application of this principle.

Thus, in addition to simplifying the geometric part of the numerical control programming problem, the systematized solution concept also greatly simplifies and clarifies the automatic programming problem as well. On the basis of these facts, the systematized solution concept is considered to be basic to any comprehensive treatment of an automatic programming problem, since this new technique provides the necessary cohesion among the various aspects of the problem. Careful observance of the principles involved makes it possible to design, and in effect build, a special computer which is particularly adapted to the problem at hand, and which has an input language which is particularly adapted to the human's view of the problem. In order to obtain solutions, then, the human needs only to express particular problems to the computer naturally and conveniently in the language. The computer can then complete all the necessary details.

I. PROTOTYPE APT SYSTEM PERFORMANCE

Before considering the long-range implications of the systematized solution concept and its application to APT system development, a brief progress report will be given on the experience to date with the prototype APT systems at M.I.T. In order to clarify the progress of the research effort, an APT system is considered to consist of a simulated APT computer connected to the human part programmer by an APT input language and to the machine tool by the necessary coded numerical language. A hierarchy of APT systems is envisioned, the levels being distinguished by the amount of automaticity involved. The APT I type of system is programmed by an APT language in terms of points in space, the APT II system is programmed in terms of space curves, the APT III system is programmed in terms of entire surface regions in space. The sequence does not extend beyond the APT III level of sophistication because it is not yet clear what the next dividing line should be.

As mentioned previously, an APT computer is actually a whole collection of simulated computers which automatically program each other in response to the English-like statements made in the APT language by the part programmer. The prototype APT systems which are now operating at M.I.T. are incomplete in the sense that the special computers which work on the human's side of the problem are not yet programmed. In other words, the APT II and APT III computers which are operating have input languages which are phrased in terms of surfaces, and therefore are considerably better than the machine tool's incremental language, but they still must be expressed in terms of numerical codes.

Initial investigations of how to design and translate English-like APT languages are currently focused on the much simpler language for the 2D-APT-II System which is being developed jointly with members of the aircraft industry, as mentioned in the introduction. Although the features of the 2-D-APT-II system will be reserved for another paper, an example of the type of APT language statement which can be "understood" by that system is

GO RGT, WITH, TL LFT, ON, CIRCLE / CENTER, PNT 3A, RADIUS, + 5.025 $$

which obviously means "Go right, with the tool on the left side, on the circle whose center is at point 3A (defined elsewhere by some other statement) and whose radius is 5.025 inches." The excessive use of commas is an artifice to make the translation job a little bit easier for the computer and will be removed in the future. The double dollar sign ($$) must be used to indicate the end of the statement, and also serves as a reminder that automatic programming, although it probably will be the most economical method for producing complex parts, is not inexpensive.

The prototype APT systems at M.I.T. do not have a convenient input language, but they do have two forms of output language. One is the punched paper tape language of the M.I.T. numerically controlled milling machine on which test parts are regularly machined from styrofoam such as is used in Christmas decorations, since it is soft so that there is no danger from programming errors. The other language is intended for human consumption and consists of pictures drawn on the output oscilloscope of the computer nearly in perspective. These pictures show the sequence of incremental tool motions as a connected chain of vectors and provide a convenient check on performance without actually using the machine tool. A number of sample pictures are shown in Figure 2; some were drawn by the APT II system with automatic indexing of curves, and others were drawn by the APT III system producing a continuous spiral path.

J. UNSOLVED PROBLEMS

The successes of the new APT system approach are intriguing and significant, but it would be inappropriate to conclude this paper with achievements which reflect only the present state of development. There still are many areas to be explored in automatic programming, since the field is so new. Consider once again the systematized solution concept which has been so central to current APT system developments. How will it fare when confronted with new, as yet unsolved problems?

The numerical control programming problem has many additional areas in which the systematized solution concept can be applied. The discovery of the solutions becomes more and more difficult as the area of application becomes more general and abstract, but adherence to the basic principles of the concept can provide valuable indications for the channeling of the research effort. An important unsolved problem of this type is that of "preprocessing" surface information from

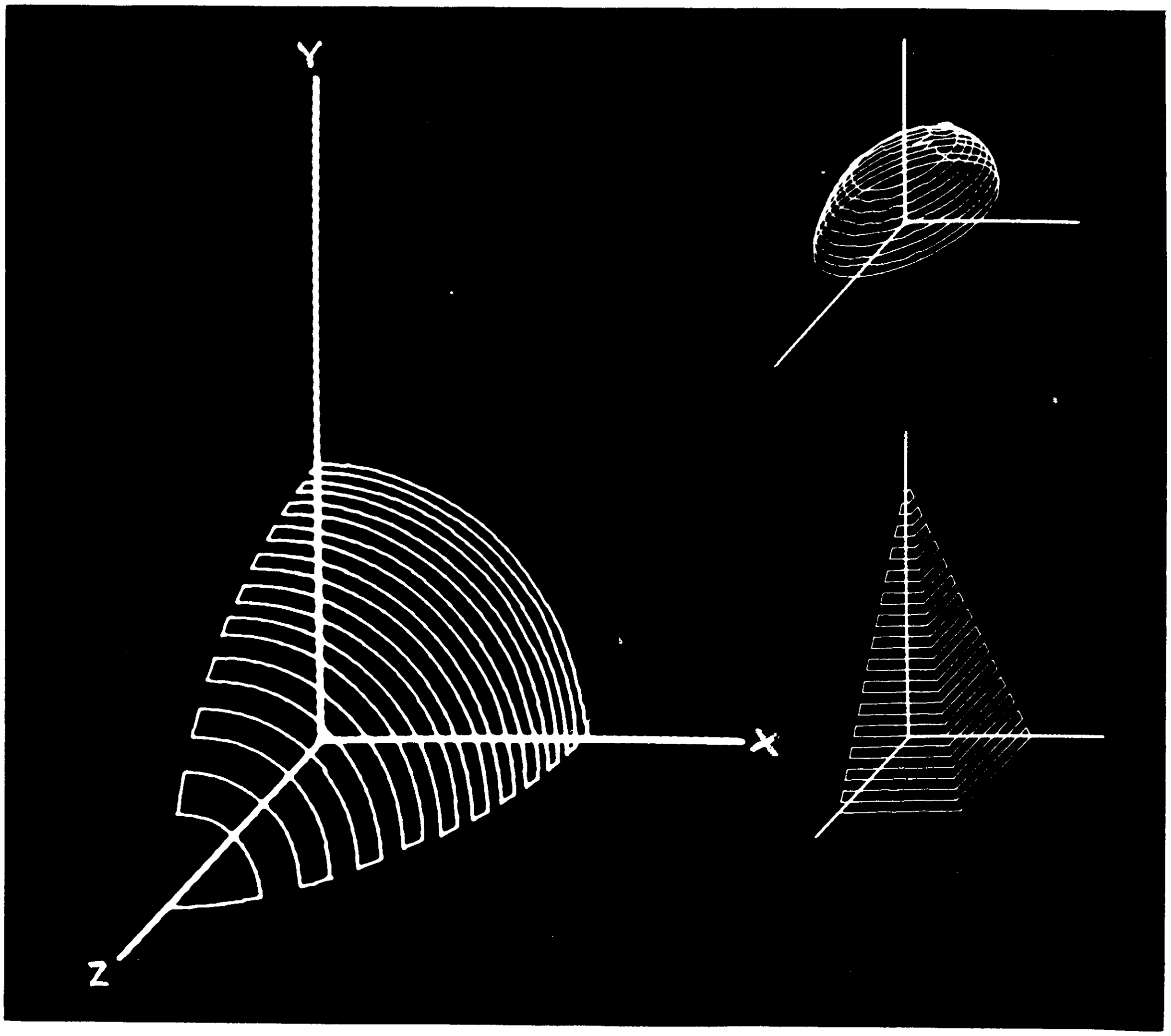


Fig. 2 Pictures Drawn by the Whirlwind Computer APT II and APT III Programs

the form which is most natural for the human part programmer, into the fixed "canonical" form which is most efficient for the calculating portions of the APT system. Although the canonical form for the sphere, for example, consists of the three coordinates of its center and the magnitude of its radius, the sphere may also be defined as being tangent to two planes passing through a point, or containing a given circle with a specified radius and center one side of the plane of the circle, etc. At the present state of development of APT systems, the preprocessing problem is being handled on a straight-forward library of subroutines basis, but clearly this is an area in which a systematized solution is possible (as indicated by the fact that the English words "passing through," "containing," "tangent-to," etc. are sufficient for defining the problem) so that much fruitful research is possible.

Another major area for research concerns the problem of translating from an English-like APT language into the appropriate behavior of the various simulated computers. In order to fulfill their mission APT languages must be real languages, with many shades of meaning and great flexibility, not merely encoding schemes in fancy trappings. There are great problems in translating powerful languages, however, since meanings of words are determined more by context than by spelling. Current research efforts are aimed at applying the principles of the systematized solution to the problem of language translation, i.e. to seek the structure of the language in such a way that correct meanings result when words are inserted. As usual, it is expected that the systematized solution should be used to simulate a special computer which is ideally suited to the problem. Although this work has just begun, there are clear indications that, by considering the problems involved in great enough depth, it will be possible to devise unusual computer types which can perform the language translation process by using the English words themselves as the input program.

The problem of designing the APT language itself requires a systematized solution approach. In order to determine what kinds of statements should be permitted it is necessary to know the structure of the part description process. How are surfaces defined? How are they related? How should tool motion be specified? These and a host of other questions can lead gradually to a realization of the important aspects of the problem, those which should be represented in the language structure. As increased complexity of machined parts is contemplated, it seems probable that future languages will not be entirely English-like, but will also employ hand-drawn sketches and symbology. At all times, the language design is aimed at increased convenience and naturalness for the human, and in order to ensure this goal, a systematized view of the role played by the human must be found.

Extension of the preprocessing and general language concepts leads directly to the idea of a "design machine" APT system in which the system assists the human in the actual design process as well as in the production of the desired part. In such a system the human would specify only the requirements which a part must satisfy, e.g. a structural member which must accept specified lateral and vertical forces and contain minimum volume commesurate with a specified amount of aerodynamic drag. Although a design machine of this type is clearly beyond present day capabilities, there are nonetheless many improtant steps which can be taken toward this goal on the basis of experience with existing systems. The problems are numerous, but the systematized solution concepts as applied in APT system design give confidence that continued developments can demonstrate the feasibility and practicality of this type of long-range thinking.

K. CONCLUDING REMARKS ON PROBLEM SOLVING

Although APT system design is most properly classified as applied research, this paper is intended to contain a message more important than a mere report of progress. In order to obtain satisfactory solutions of the kinds of problems which arise in these studies, it is necessary to delve deeply into basic questions concerning how to solve problems. The term "systematized solution" is not very descriptive, but the concept referred to is nonetheless a valid one, and one of extremely wide generality and power. It seems that the reason that the systematized solution concept can be applied to such diverse problem areas as geometry, language design, and language translation is because it really represents an imperfect formulation of a fundamental truth about problem-solving. At the present time the full implications of the concept are not clear, and any attempts to be more explicit in expressing the principles involved are doomed to appear trite. The main difficulty seems to be the lack of a rigorous definition of just what constitutes a "problem." It is expected that further clarification will result from further studies in automatic programming and related topics. It is hoped that others will attempt to find systematized solutions for problems in their own fields as well.

Many times in the past, technological advances have stimulated new basic scientific research. The advent of the programmed machine for the first time permits abstract knowledge to be applied directly to physical processes. Perhaps new insight into the way human beings solve problems will evolve from attempts to make machines to help them.

APPENDIX

EXAMPLE OF SYSTEMATIZED SOLUTION

Since it is difficult to convey accurately the real meaning of the term "systematized solution," a simple illustrative example is given which demonstrates not only the idea, but also the type of reasoning which is used in the development of systematized solutions. Consider the problem of calculating the point of intersection of a parabola and a circle. It is possible to use analytic geometry to develop equations which give the solution directly, for any parabola and any circle. The method is rather complicated, however, and this work may be avoided by making a computer program which "looks" for the intersection using a trial and error search. Because of the high speed of modern computers, the searching method is economical, and may even be more efficient than the straight forward analytic solution.

For the purposes of this discussion, it may be assumed that the computer program starts with a point at the "top" of the parabola and that the direction to move along the parabola to find the desired intersection point has been determined. (See Fig. 3) After moving to a new point on the parabola, the program computes the distance from that point to the center of the circle. At the desired intersection

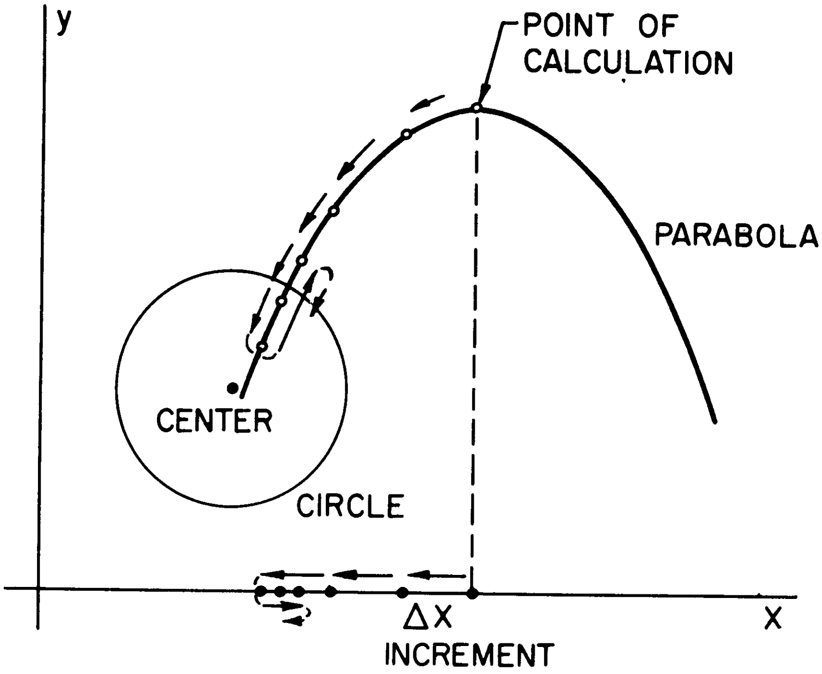


Fig. 3 Method for Finding Intersection

point the distance will equal, within a tolerance, the radius of the circle so that the intersection point may be detected. If the current point of intersection is not close enough to the intersection point, another "step" along the parabola is taken and the test is repeated.

Because the program steps along the parabola by a fixed amount each time, it may pass by the intersection point without getting close enough to be within the tolerance. In this case, the program should reverse direction, taking smaller steps. The fact that the intersection point has been passed may be detected by comparing each error with the preceding one. As long as the error is decreasing, the program continues in the same direction, but as soon as the error increases the program reverses direction and takes steps half as big as the preceding steps. In this manner, the program eventually will reach a point which is within the tolerance and which will then be called the intersection point. A simple flow diagram of the completed program is shown in Fig. 4. The operation of the program as shown in the flow diagram, is indicated by the descriptive title given each step.

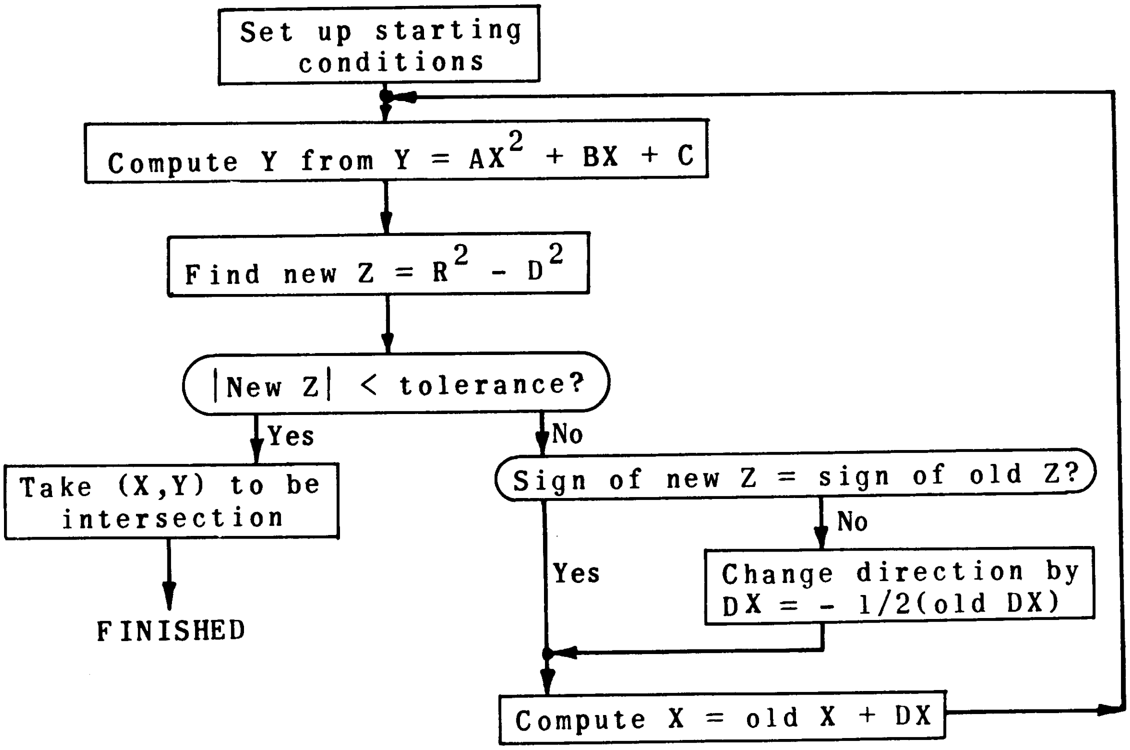


Fig. 4 Flow Diagram to Find Intersection of Parabola and Circle

The flow diagram of Fig. 4 cannot yet properly be called a systematized solution since it only solves for the intersection between a parabola and a circle. Notice, however, that the box which says "Compute Y for $Y = AX^2 + BX + C$" may actually be replaced by a box which says "Compute Y from $F_1 (X,Y) = O$" without in any way altering the other boxes in the flow diagram. Thus, the choice of a mechanistic step-by-step solution to the circle-parabola problem instead of the analytic approach has led immediately to a systematized solution for finding the intersection of any curve with a circle. Notice that all of the hard part of the problem is contained in the basic method itself which is represented by the flow diagram, and in order to solve any particular problem it is necessary only to substitute the function F_1 for the particular curve desired.

By using one other mathematical fact, the systematized solution above can be made even more general. For any mathematical expression of the form $F(X,Y) = 0$, a surface in three dimensions is obtained if $F(X,Y)$ is set equal to another variable Z. This surface will intersect the X,Y plane in the curve $F(X,Y) = 0$, and for any point (X,Y) <u>not</u> on that curve, the value of Z will be positive or negative depending upon whether the surface is above or below the X,Y plane in that region. (See Fig. 5) Notice that if Z is positive for one point in a region of the X,Y plane, it will also be positive for all other points in that region. Except for unusual surfaces Z will be negative for points on the other side of the dividing curve $F(X,Y) = 0$. Therefore, it is possible to tell whether a point is on one side or the other of the curve, $F(X,Y) = 0$ by the sign of Z. This property may be used to replace the box of the flow diagram which refers to the circle, since it is only necessary to know when to reverse direction and take smaller steps.

Without further changes in the flow diagram, the coordinates of the point which has been determined to lie on the first curve may be taken and substituted into the expression of the other curve. The resulting value of Z indicates which side of the intersection point the current point of calculation is on. The revised flow diagram is shown in Fig. 6 and represents a systematized solution to the problem "Find the intersection of any two plane curves." The program itself is concerned only with the problem of intersection and is not concerned in any way with which two curves are being used. The only requirement to solve any particular intersection problem is to substitute the appropriate programs for the particular curves.

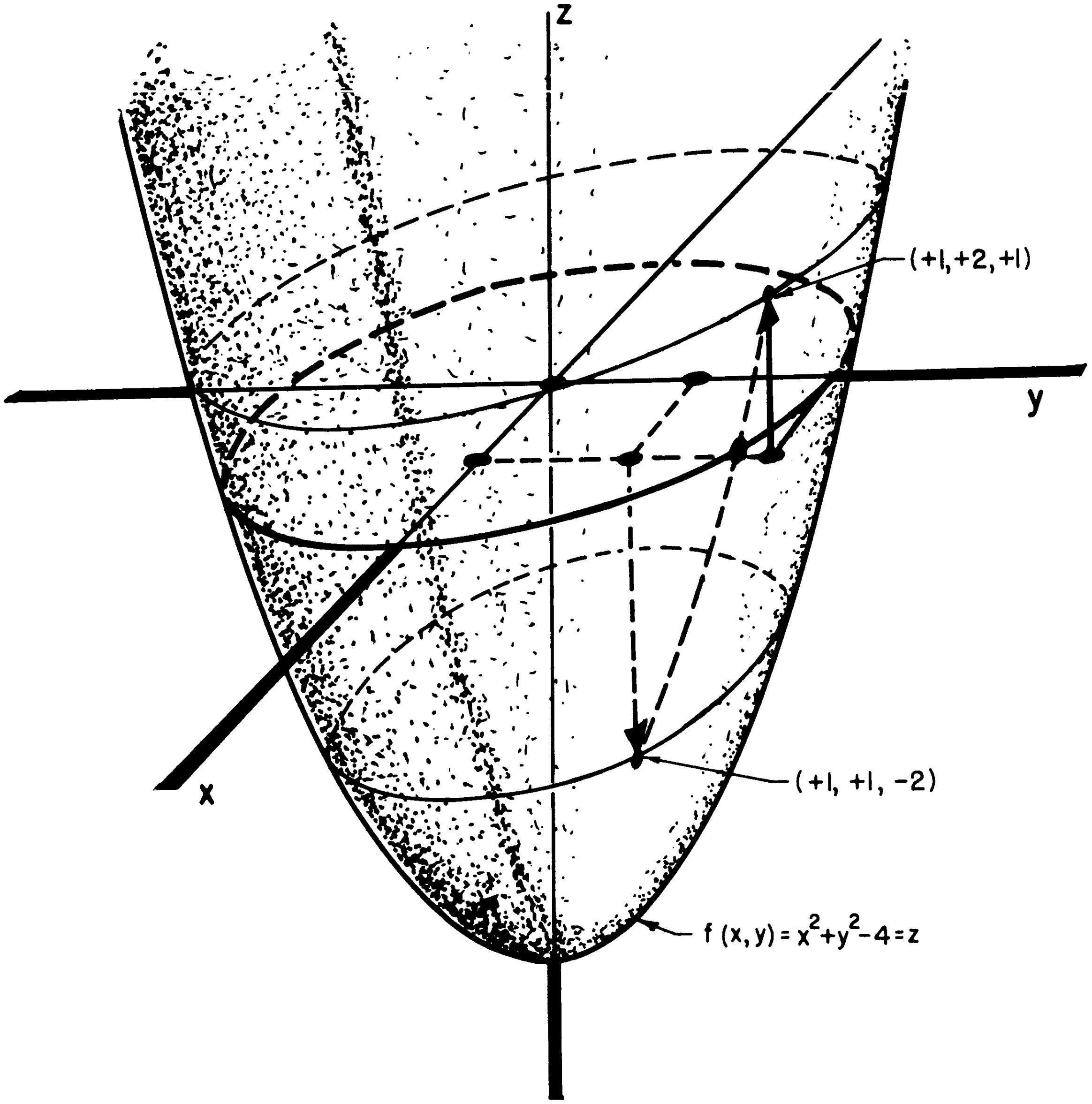


Fig. 5 Construction for Determining the Inside or Outside of $F(X, Y) = X^2 + Y^2 - 4 = 0$

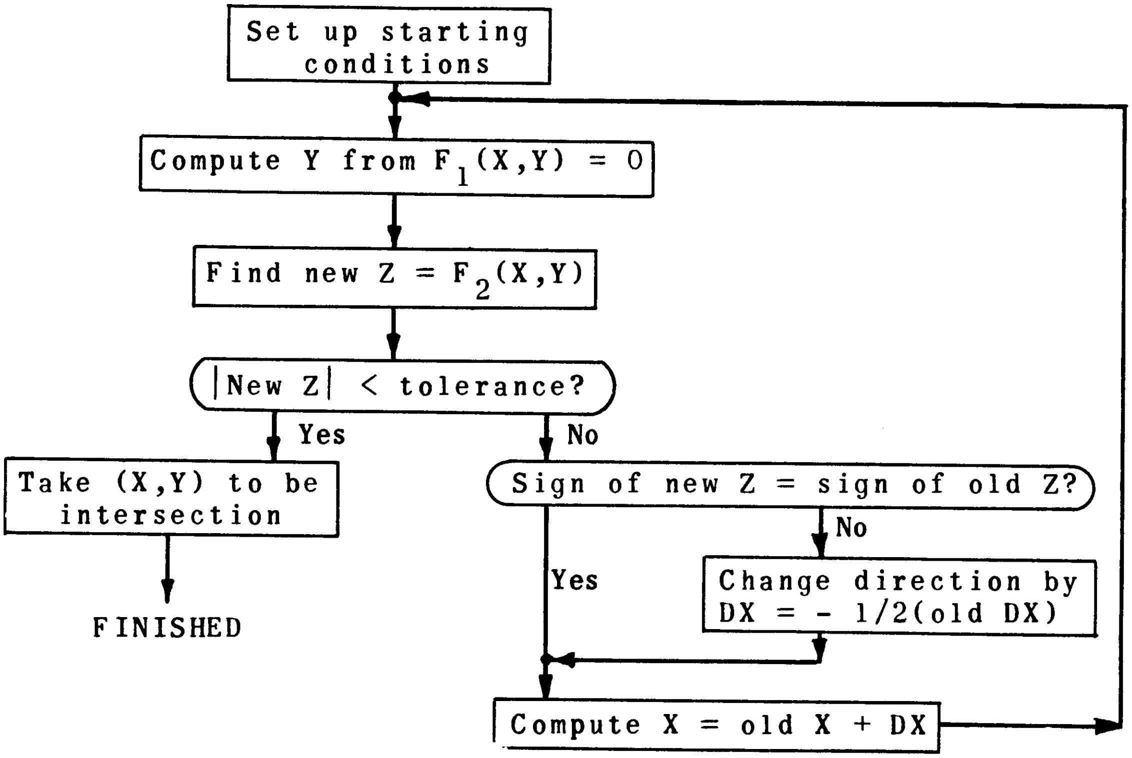


Fig. 6 Flow Diagram to Find Intersection of Two Arbitrary Plane Curves