A MINICOMPUTER PREPROCESSOR AND SYSTEM MACROS FOR THE APT LANGUAGE

by

Ben De Smit, Dipl. Eng.

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Master of Engineering

McMaster University
December 1977

TO MY PARENTS

Master of Engineering (1977) McMaster University

(Mechanical Engineering) Hamilton, Ontario

TITLE : A MINICOMPUTER PREPROCESSOR AND SYSTEM MACROS FOR THE APT LANGUAGE

AUTHOR : Ben De Smit, Dipl. Eng. (Katholieke Universiteit van Leuven, Louvain, Belgium)

SUPERVISORS : Dr. I.Yellowley

Dr. H. El Maraghy

NUMBER OF PAGES : vii, 116, Appendices, 95.

ABSTRACT

In this work a new NC design, based on the existing APT system (processor and postprocessors) and consisting of an additional front-end interactive Basic preprocessor, is presented. Software was developed that enables a part programmer to generate a part program on a 16 K minicomputer (PDP11/34). In order to fully exploit the off-line capabilities of an intelligent terminal, routines were written that display the geometrical input. While verifying the latter the user is offered the opportunity to instantaneously modify the displayed geometry if required.

Subroutines of the APT part programming language (macros) were developed to cover basic geometrical configurations of turned parts (bar stock and forgings) and 2½ axes milled pockets. This approach reduces part programming to inserting dimensions of the workpiece and the technology involved.

As far as technology is concerned, a milling optimizaroutine was developed that can direct the user in the selection of a cutterradius, feed and rotational speed when machining pockets.

ACKNOWLEDGEMENT

I would like to express my deep personal appreciation to Dr. I. Yellowley for his invaluable guidance and encouragement and his perceptive direction in this work. Sincere thanks to Dr. H. El Maraghy for her interest, her helpful advice and her active co-supervision.

I have been fortunate in having an informal association with the Canadian Institute of Metalworking and in this regard the cooperation and the encouragement of Mr. C.J.Gouldson, Mr. J.E.Grozier and Mr. T.G.Butlin are gratefully acknowledged.

The financial assistance in the form of a teaching assistentship in the Department of Mechanical Engineering is also acknowledged.

Finally I would sincerely like to thank Linda for her excellent typing. Her patience and guidance were a significant factor in the succesful completion of this work.

TABLE OF CONTENTS

CHAPTER 1.

Introduction.

1.1. Introduction to numerical control. [1,2,3,4,5]

Machine tools are one of the most essential items in our society. The phrase : "The quantity and quality of a nation's machine - tool industry is in direct proportion to that nation's place in the world - pecking order." [1] could not have accentuated the idea behind numerically controlled machine - tools any better. Besides rapid set - up times and greatly reduced leadtimes, NC equipment also gives superior accuracy and repeatability.

In 1963 the total number of numerically controlled machine - tools installed in the US metalworking plants did not exceed a few hundred. In 1968 the US government statistics show 14,000 numerically controlled machine - tools on the job. In 1973 the American Machinist's Inventory of metalworking equipment reported nearly 30,000 items of numerical control equipment in place at that time. A rough estimate of the total amount of NC machine - tools installed in North America today, would be 40.000. The same trend prevails in other major industrial areas, such as Europe and Japan.

note : numbers in between square brackets indicate reference - works (see references - listing).

For nearly two decades numerically controlled machine tools have become increasingly important in the North American metalworking industry. Numerically controlled machine tools can be programmed manually or by using digital computer programs. Manual programming is carried out completely by hand using basic rules of mathematics. In producing the tape the manual programmer interprets the design and converts its data into machining instructions. He ought to be familiar with the machine tool and the control unit. The manual programmer not only calculates the cutter path but he also states the functions required for each machining operation (e.g. toolfunctions) and determines spindle speeds and feedrates. Obviously, this type of programming is tedious, timeconsuming and prone to errors.

In order to reduce the time for programming, a digital computer program can be used to carry out the calculations and format manipulations. So in the last decade a great deal of effort has been expended in the development of computer-aided programming methods in order to make the most efficient use of this highly efficient and expensive machinery.

Initially these efforts were quite expensive as manpower was concerned, and so their applications were limited to the aerospace industry. But in recent years the concept of computer-aided programming has spread very ra-

pidly to the rest of industry and has shown to be successful. The term "part-programming" means nothing else but inputting information to a digital computer program (e.g. APT)*. This is done by writting a manuscript (called part-program) in a kind of pidgin english which is closer to the programmer's own than is the basic machine language coding (a good comparison would be Fortran as opposed to Assembly language). This input defines the geometrical shape to be produced and also the method by which the cutter will operate.

There are two broad categories of numerical control languages, machine-oriented languages and general-purpose languages. Machine-oriented languages, as the name implies, are written to produce numerical control tapes that can be used for one specific combination of machine and controller (e.g. SPLIT and SNAP). This type of languages takes into account the functions and limitations which that particular machine is expected to perform.

The general-purpose languages are designed to be more universal. They describe programs to be used by any of a large number of diversified numerically controlled machines. First, a generalized program (called "processor program") compiles the part program and utilizes its infor-

* APT is an acronym for Automatic Programmed Tool, the most widely used numerical control system.

mation in order to produce the coordinates of the cutter centre path. However, it will not take into account any machine characteristics. Finally, the output from the processor is used as input for a further program, called "postprocessor program", of which the output can be utilized to control the machine tool. It takes into account the specific characteristics of the machine tool/control system, which the processor does not. An example of a general-purpose computer program (language) is Automatic Programmed Tools (APT). APT is not only the most widely used of the computer languages for numerical control but also the most powerful as it is capable of automatically manipulating, transforming and scaling geometry and motion statements.

Part programming is significantly simplified with computerized numerical control systems, where a self-contained general-purpose digital computer with sophisticated software, such as APT, can relieve the part-programmer of much of his tedious calculations.

1.2. State of the Art. [2,3,6,7,8,9,10,11,12,37]

Up to this date the majority of the installed machine-tools is still programmed manually (i.e. an errorprone and timeconsuming process). In paper [2] from TNO Metaalinstituut, it was estimated that the total demand for control programs approximated 2,000,000 programs/year, which on the average would result in 5000 manyears/year of actual manual programming effort. This illustrative example emphasizes the cumbersome, laborous and timeconsuming aspect of manual programming. A survey, carried out by TNO Metaalinstituut (Netherlands) in 1975, revealed a 1/10 timeratio of part programming when using their "on-line" MITURN turning-part-programming system versus manual part programming for an average 140-block-tape. It should be noticed that the above mentioned timeratio drastically increases for complicated parts. Indeed, a more general survey, including various languages and multiaxes programming, was executed by the Numerical Control Society (Michigan, USA) and showed an average 1/20 ratio. [3]

In many technical and commercial fields, the digital computer has provided a powerful tool for solving problems. For the last two decades, its calculation and manipulation power has been utilized increasingly by various industries.

Traditionally the digital computer has found its application in several aspects of manufacturing. In the process industries the digital computer is often in charge of control and data acquisition. In a large number of firms the digital computer has already been applied to production and inventory control.

More recently attention has been focussed on the automation of part programming production for NC machines.

Eventhough NC devices are now becoming very common in metalworking industry, the application of computer-aided manufacturing has been slow. The demand for computing equipment often originates in the financial department of a company. An interesting study by the NCS shows that many firms are already limited to the computer they are presently using and must find a software package that can be run on the in-house machine. The survey [3] revealed that manufacturing managers are not sufficiently represented at the time their firm chooses its computer system and it also discloses that only 49% of the NC managers had been adequately consulted about their computational needs in present and future. Such a neglect reflects an obvious lack of communication between the people who control computer operations and the manufacturing managers. Too often neither group is aware of the possibilities the computer can offer manufacturing departments. Fortunately, during the last decade a lot of work has been performed successfully to render NC systems less computer-dependent. Especially the APT* system (its general processor has been written in Fortran IV) allows easy interchangebility among most computers, which more or less solves the above mentioned problem.

In contrast with other computer languages such as Fortran, Basic and Cobol, the numerical control languages such as APT, ADAPT, etc... are used specifically to produce numerical control tapes. Numerical control languages are at a most sophi-

* see further

sticated level as shown in Fig. 1.1. Indeed, the NC languages require the greatest degree of compilation (translation) by the computer before they actually can be processed at the binary level. As mentioned above, already some NC operating programs (processors) have been written in Fortran IV. Languages, such as APT IV, must first be translated by a compiler to Fortran IV, then reduced to an assembly language and then to binary code.

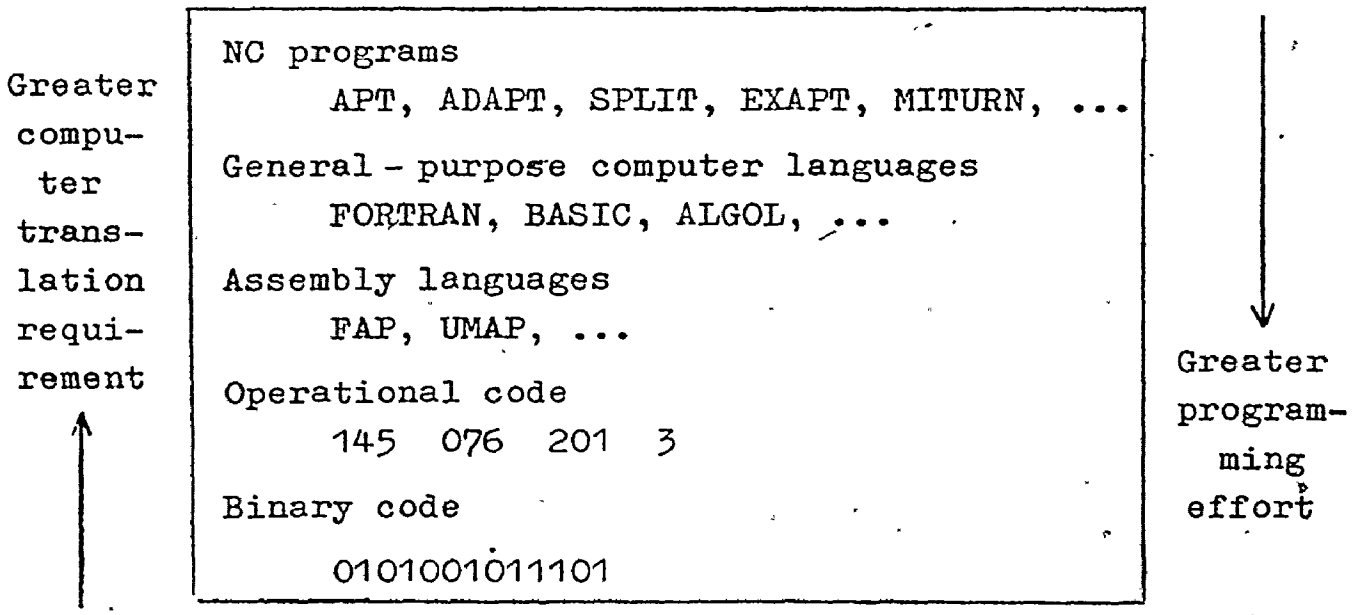


Fig. 1.1. Hierarchy of computer languages

In 1972 the NC lathe computer programming languages have been studied and classified by J. Tlusty and G. Dunsford [7]. They based their classification of NC lathe languages on the level of automation as suggested by Prof. J. Pétgrs. The languages can be divided into three groups according to their automation level. An appropriate definition of the automation level of a language would be the number of operations that language integrates in

the program. The operations pertain to items such as the determination of tools, cutting conditions (more general "technology"), etc ...

In the first group the computer program provides the cutter location data (commonly referred to as CL data), using as input data only geometric definitions of the toolpath. This group includes APT and some APT-like languages (e.g. ADAPT,...). APT was the first computer-assisted numerical control language to become available to numerical control users. In 1956 [8], the first version of APT was produced at M.I.T., as a result of an overall project for the development of numerical control, largely sponsered by the US airforce. The APT system was further refined and developed at I.I.T. Research Institute (via a project called APT Long Range program) while financed by corporations and government agencies in the US, Canada and Western Europe. APT is not only a powerful system for producing tapes for numerically controlled machine tools, it is also a part programming language, capable of performing all the arithmetic and geometric calculations, in order to describe parts, specify cutter motions and machinetool functions. A more thorough analysis of the APT system will be dealt with in chapter 3.1.

In the second level of automation the geometric definitions of raw material and finished part are sufficient for the computer to provide the cutterlocation data. Indeed, the computer subdivides the material to be removed into successive passes. Programming in a language such as APT often is timeconsuming because of the lengthy input. This defficiency led to the development of a second group of languages (e.g. COMPACT II,

CINTURN*, ...). These languages, based on "canned - cycles"**, let the computer rather than the part programmer decide how operations should be carried out. Compared to group I languages, the input is obviously reduced.

COMPACT II, now capable of 5 - axis - simultaneous motion***, has become a powerful and commonly - used numerical control language in North America. Unlike some numerical control languages (e.g. SPLIT****, ACTION II, etc...) it is not restricted to a certain manufacturer's machine - tool and thus applicable to all numerical control machine - tools. Eventhough the part programmer still has to input toolradius, cutting speed and feed, maximum depth of cut and finishing allowance, COMPACT II offers a series of systemmacros (containing various cutting sequences) available to every programmer. So far this flexibility is practically (except for one simplified pocket-routine) non existant in APT.

In the third level of automation the computer integrates into the cutterlocation programs all technological command data about speeds, feeds, canned - cycles, etc... Not only is the part programmer relieved from his task of determining the tool-

* CINTURN is actually not a NC language, but rather a collection of macros.

** Canned - cycles are special computerroutines designed to generate a whole series of operations from a single command.

*** It needs to be emphasised that 5 - axis - simultaneous positioning is meant and not 5 - axis - simultaneous contouring as is the case with APT.

**** SPLIT : Sundstrand Processing Language Internally Translated.

paths (like in group II languages), but also the "technological parameters" are calculated by the computer. Indeed, analoguous to group II, the input data consist of geometrical definitions of the initial and final profile, but additional material, tooling and machine characteristics will enable the computer to calculate the cutting conditions (technology). In most cases the computer is attached to on-line libraries which contain the various material, tool and machine data-files.

An example of this group of NC systems and meanwhile a true example of the cooperation in European numerical control research and development, is the EXAPT system. The MITURN/GE-TURN system, also originally European, belongs also to this (highest) level of automation [37].

Historically, the trend in the usage of APT has been the tailoring of a broad flexible language to specific needs. This general approach has been amplified by the development of languages based on APT, notably EXAPT. The EXAPT system, developed at the technical institutes of four major German universities, consists of three languages : EXAPT 1 is a point-to-point and straight path language, EXAPT 2 is for programming turning operations, and EXAPT 3 is a further extension of EXAPT 1 and allows the programming of contour milling operations. EXAPT 2 attempts to generate all the toolmotions from a simple description of the blank and a finished part description. Of these three languages, EXAPT 2 is the most widely used, while EXAPT 3 is still in the research and development stage. [9]

Fig. 1.2. shows the principal procedure for programming and processing an EXAPT program in a computer. Based on data

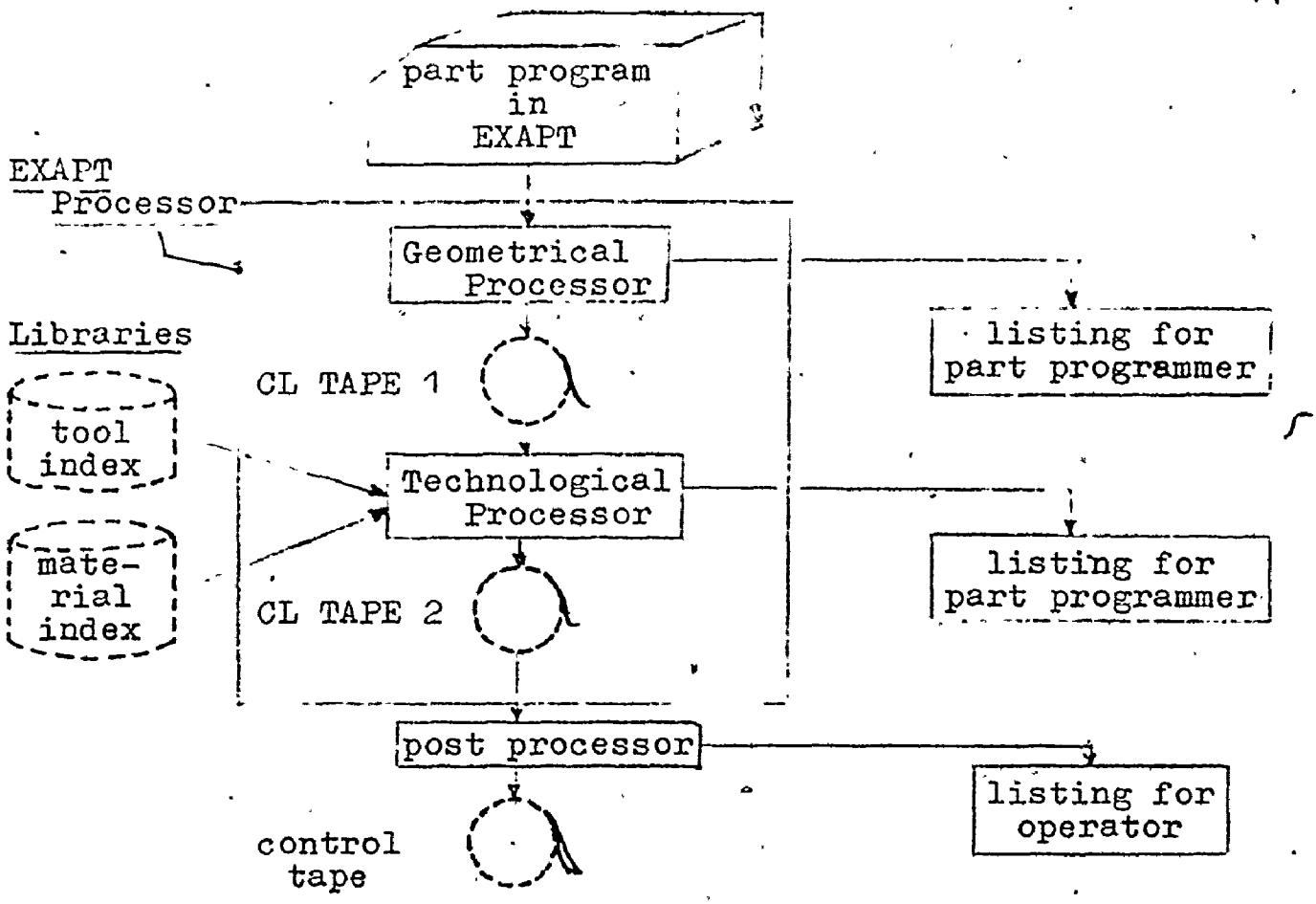


Fig. 1.2. Production of a control tape via EXAPT 2.

given by the workpiece drawing (blueprint), the part programmer describes the workpiece in an APT - like language. This part program is processed in two sections. In the first phase, the general geometric and technological processing are carried out. The technological data are determined with the aid of the tool- and materialindeces (or files). Then the program is processed in the postprocessor in order to adapt the output to the requirements of a specific NC machine - tool. The processing in the postprocessor produces the punched tape for the control of the machine and an output - listing for the operator of the particular machine - tool.

The current status of the technological routines within EXAPT 2 is not known by the author at the present time, however,

the amount of usage of the full system on complex parts has been very limited. The research and development on EXAPT 3 started in 1967 and has, to the author's knowledge, not yet been industrially applied and utilized at this point in time. A recent EXAPT report (minutes 1976) [11] states that the work on EXAPT 3 has been terminated and soon will be published, as is.

The MITURN programming system (Metaalinstituut Turning program) was developed by the Centrum voor Metaalbewerking van het Metaalinstituut TNO of the Netherlands. At the present, the General Electric Company has taken over MITURN and offers it on the General Electric Mark II Time - sharing network as GETURN. The present version of MI/GETURN has been released in March 1973 and is being distributed commercially to customers in North America, Europe and Japan through a worldwide Time - sharing system, with an intensity of usage of respectively 50, 41 and 9 percent. [6]

The fairly new MI/GETURN system is specifically designed for lathes. This self - contained programming system is in fact a comprehensive production system composed of several subsys - tems which together make an important contribution to the op - timum utilization of numerically controlled lathes. It is equip- ped with both geometrical and technological subsystems which re- sult in a high degree of automation, for instance, in GETURN :

- the optimum sequence of operations is determined
- the optimum depth of cut, feed and cutting speed are determined per cut
- etc...

In order to carry out the above mentioned functions, GETURN is based on group technology for its input files. These files consist of a tool file, a methods file, a material file and a machine file. These files stay fairly constant for a long period of time but need occasionally be updated. The relationship between the general input and output of the GETURN language is depicted in figure 1.3.A.

The programming effort (time + money), while using the EXAPT system, becomes, comparably, as low as special "group technology" systems. However, the EXAPT system has greater flexibility and range of application. Furthermore, the modular structure of the system is the precondition for constant adaptation to further developments of the NC technique. A comparison between EXAPT and a highly automated and sophisticated "group technology" system is depicted in Fig. 1.3.B. [10]

Eventhough the latest figures show that manual programming is still in use for about 60 - 70 %, the number of different programming systems for NC machine - tools has proliferated. In 1976, a TNO paper [12] estimated the number of existing NC programming systems to be about 150. Eventhough the analysis of this vast array of developments is obviously outside the scope of this work, the author will discuss a few approaches to interactive part programming aswell as review some integrated CAD/CAM systems in chapter 3.4.

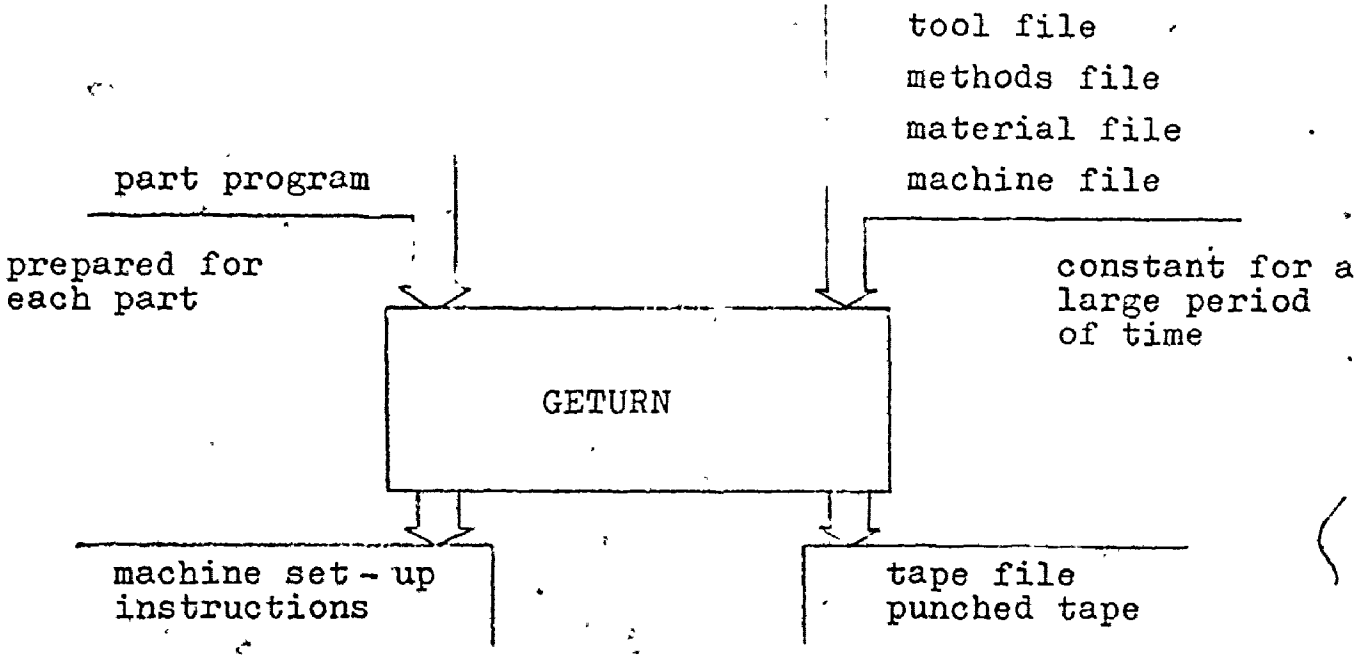


Fig. 1.3.A.

Relationship between the general input and output of the GETURN language.

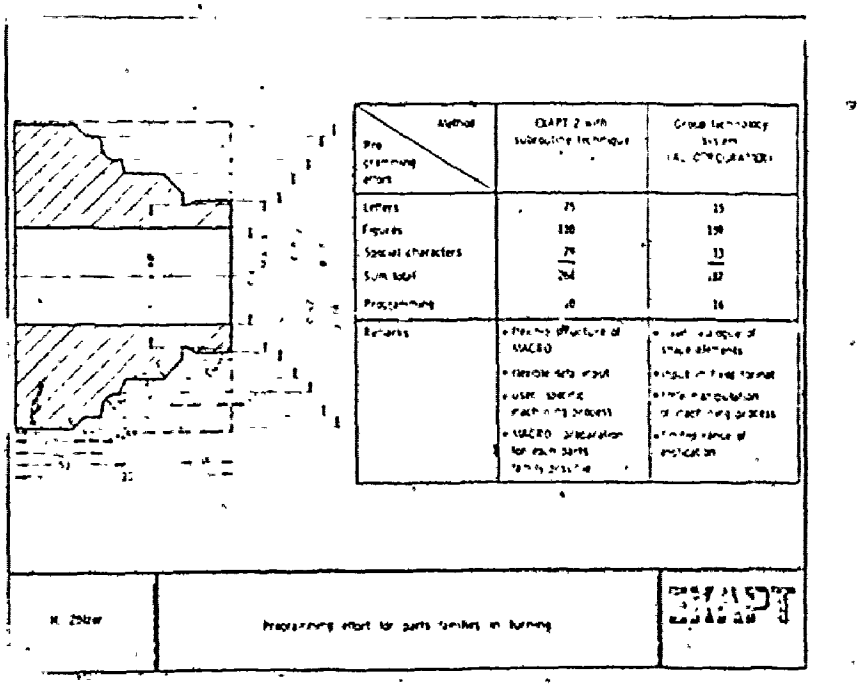


Fig. 1.3.B.

1.3. Aims of this work. [1,10,13,14,15]

An authentic example, in which a company had to decide upon which NC system to choose, might be an interesting means of introducing the aims of this work and the new design involved. A recent report [13] in the EXAPT Minutes 1975 on the introduction and application of the EXAPT system in a middle-sized German firm shows an interesting economical (financial) comparison between two numerical control systems. Because of several reasons (including limited computer capacity at a high requirement of paper tapes), the firm decided against minicomputerlanguages. One of the reasons that their choice (between APT and EXAPT) fell on EXAPT was that the latter system seemed free of mathematical computation. The comparison between APT and EXAPT 2 resulted as follows (Table 1.1.) :

Designation	units	APT	EXAPT 2
Number of statements in the part program		166	86
Preparation of the part program	min	300	160
	%	100	53
Costs for the program	DM*	180	160

Table 1.1.

* DM : Deutsche Mark
(West German monetary unit)

The following statement was made by the staff of the firm's manufacturing departement : "The reduction of time for the preparation of the part programs was of decisive importance to us.". This phrase clearly depicts the disadvantage of the less - automated APT system. Indeed, compared to the EXAPT system, it took twice as long to prepare an APT part program.

However, when comparing both costs, one can obviously notice the slight price difference (180 DM when using APT and 160 DM when employing EXAPT). Bearing in mind this almost non-significant price - difference, it might even have been the company's affinity for a German NC system that inclined the balance. Anticipating probable future expansions of the company's manufacturing departement, it needs to be stated that the APT postprocessors are far less expensive than the EXAPT post-processor. The new NC design, presented in this thesis, is based on the APT system and thus has intrinsically all the APT system advantages. It meanwhile possesses the high degree of automation peculiar to EXAPT. In the above mentioned table (Table 1.1.), the number of statements aswell as the total time of part program preparation would be approximately reduced by a factor of 2 (thus comparable in time with an EXAPT approach). Needless to say that their would be a fair reduction in cost involved too !

When compared to the ambitious aims laid out in a multitude of research papers (e.g. TIPS 1 [14], Volume - building - brick system [15], ...) it would seem that the work described here is rather mundane and may be behind the times as its fundamental is one of the oldest NC systems : APT. It should, how-

ever, be pointed out that tremendous economic gains are available when slightly modifying the current existing APT system.

Our first aim is to upgrade the APT system to a semi-automatic programming system without loosing the power and versatility APT is known for. To date, APT is probably not only the most powerful but also the most widely used NC language in North America. An illustration of its popularity is the fact that in 1971 the Army Material Command of the US Army [1] selected APT as the standard NC language because the NC field within the US Army had created a "Tower of Babel" situation which disabled almost all communication among several machine shops and which inevitably brought down the overall efficiency. Not only is the APT system machine-independent, but its processor and postprocessor, written in Fortran IV, have been implemented on more computers than any other language. Indeed, APT is also advantageous in terms of postprocessing. At present, machinetool users often desire from the manufacturers the necessary postprocessor for the programming language they are interested in. The predominant role of APT in the North American NC scene makes the APT postprocessor readily available for almost any NC machinetool with a price remarkably lower compared to that of postprocessors of other NC systems. However, in contrast with group II and group III languages, APT is not capable of automatically determining the various cutting parameters or optimum sequence of cut, nor does it have the capability of relieving the part programmer from the dull, repetitive work of defining all the subsequent cuts, needed to machine a part. The capability of determining "workshoptechnology", such as tooling,

feeds, speeds, ... and the provision of canned cycles (macros) for automatically defining the operation sequences and tool-paths to machine a part would turn APT into a superior system.

Our second objective is simple : to save money by reducing part programming costs. Inputting data to a preprocessor can be considerately shorter, and requires substantially less thought to write than the equivalent full part program. This method allows computers to do what they do best and, at the same time, it leaves the programmer or engineer free to be creative. With this tool the programmer should be more effective in carrying out his day - to - day task. Indeed, for a relative small financial effort a part program can be prepared on a minicomputer. A BASIC preprocessor, that can handle both a non - graphic and an interactive graphic front - end input format, would provide the existing APT system with a conversational and technological module. Moreover, besides editing capabilities and a proportionate degree of diagnostical analysis to spot the detectable errors, the preprocessor includes a graphical part geometry verification module.

While applying this NC design, the user should have a system at his disposal that could cover a large range of parts in the 2½ D domain. Indeed, the list of macros, developed up to this stage, is not complete, but the goal of this NC design is not having the almighty capability of machining every existing complex shape. According to a recent German paper ("Market Situation for EXAPT" out of "Minutes 1975") [10], the authors claim that the EXAPT system, also limited to machining tasks up to 2½ D, can deal with more than 90 % of all NC machi-

ning tasks in the Federal Republic of Germany. The reader should bear in mind, however, that by "disconnecting" the preprocessor from the front-end, the user would still be capable of part programming complex shapes, utilizing APT's full power (> 2½ D).

Recently the APT programming system has often been implemented in central computer centers and made available practically everywhere through public telecommunication lines (e.g. acoustic couplers). Not only does the simple and versatile accessability bring the use of large computers containing powerful programming systems to small plants but, in contrast with the batch processing mode, the turn-around time in between resubmissions is drastically reduced. In this environment our system can be depicted as an intelligent terminal backing up the existing APT system by offering the user the following extended off-line capabilities :

- interactivelly inputting the necessary information for part programming preparation
- part geometry verification on CRT display
- the provision of metalcutting technology and optimization routines
- editing of system macros in particular and part programs in general

A simplified view of the procedure is depicted in Fig. 1.4. It shows the links between the various concepts in the system.

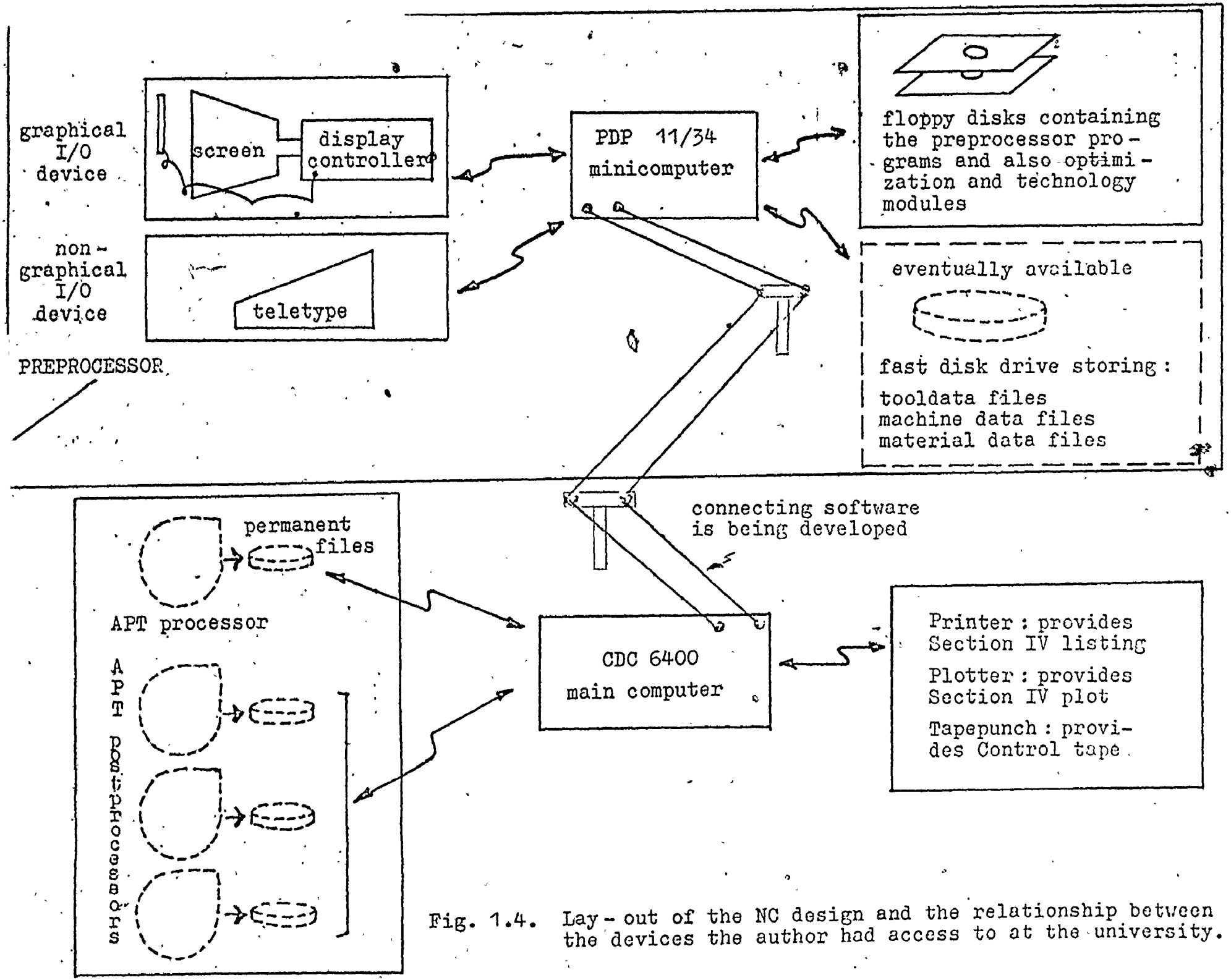


Fig. 1.4. Lay-out of the NC design and the relationship between the devices the author had access to at the university.

CHAPTER 2

The selection of cutterradius, feed and circumferential velocity in peripheral milling.

2.1. Introduction.

As far as the semi-automatical stipulation of technological parameters during the preprocessor phase of this NC design is concerned, so far only the selection of economic machining conditions in peripheral milling has been implemented. However, parallel to this work, the optimization of multipass turning is being undertaken at the university. This knowledge can later be integrated in the preprocessor software package.

The selection of machining conditions in peripheral milling was dealt with in [19] and [20]. The implementation of the concepts, developed in these research papers, is undoubtately of practical significance in the development of technological routines for NC programming. Both papers discuss the stipulation of feedrate and velocity for one milling cutter when radial width and axial depth of cut are varied in a roughing operation. Before presenting the approach to the selection of machining conditions in a pocketing operation, it is necessary to briefly explain the concepts underlying the development of a toollife equation for phcripheral milling as presented in the above mentioned papers.

The inherent complexity of peripheral milling in comparison with turning results in an additional number of practical variables when assessing toollife. Besides being discontinuous which leads to a possibility of both thermal and mechanical

shock, the process is characterized by a variable chip thickness during cut.

In [16] the author, assuming a linear relation between flankwear and cutter-time and basing himself on the chip thickness (S) - toothlife (T) relationship

$$(S + A) \cdot T^{\beta'} = \text{const.} \qquad (2.1)$$

with A and β' constants

as proposed by Kronenberg, created the concept of equivalent feedrate (S_{eq}) which, when maintained constant, would yield the same average wearrate as the variable chip thickness in milling :

$$S_{eq} = \left[\frac{1}{\phi_s} \int_0^{\phi_s} (S + A)^{1/\beta'} \, d\phi \right]^{\beta'} - A \qquad (2.2)$$

where ϕ_s is the swept angle of cut.

In case of a "Taylor type" relationship ($ST^{\beta} = \text{const.}$), S_{eq} is given as

$$S_{eq} = \left[\frac{1}{\phi_s} \int_0^{\phi_s} S^{1/\beta} \, d\phi \right]^{\beta} \qquad (2.3)$$

S_{eq} expresses the influence of cutterdiameter, width of cut and feed per tooth on the geometry of the cut.

The influence of the intermittent nature of the process on toollife, predominantly thermal stressing, can be expressed by the thermal fatigue parameter, introduced by Yellowley in [16] as

$$[X] = [E_R] \cdot [N \cdot x]^{\frac{1}{2}} \qquad (2.4)$$

where E_R is the range of thermal strain for an arbitrary rake face temperature of 100

N is the cutter rotational speed (rpm)

x is the ratio of total cycletime to the time in cut during the cycle

The thermal fatigue parameter takes both the range of thermal strain and number of cycles of thermal strain per unit time into account. E_R characterizes the relationship between heating time, cooling time and range of thermal strain. In [17], the following expression was suggested for E_R (see Fig. 2.1.) :

$$E_R = 39 \log t_c - 23 \log t_h + 37.5 \qquad (2.5.)$$

where t_c is the cooling time (msec)
t_h is the heating time (msec)

The E_R values show very little difference for high speed steels and carbide tool materials (for the same heating and cooling times).

Assuming a "Taylor type" relationship between active toollife and velocity and taking into account the intermittent nature of the process, the following active toollife (T_a) equation has been proposed in [19] and [20]

$$T_a = \frac{\text{const.}}{[X]^m \, [S_{eq}]^n \, [V]^p} \qquad (2.6)$$

and consequently the actual toollife of the cutter (T) is

$$T = \frac{360}{\phi_s} \frac{\text{const.}}{[X]^m \, [S_{eq}]^n \, [V]^p} \qquad (2.7)$$

where ϕ_s is the swept angle of cut and V the circumferential velocity. Typical exponent values for toollife equations (2.6) and (2.7) are $m = 2$, $n = 1$ and $p = 2$. These toollife equations will be used in further investigations.

2.2. Optimization of peripheral milling.

2.2.1. General.

Based on the above mentioned papers [18], [19], [20] as well as [16] and after an initial study of the problem carried out on the CDC 6400 using the 3D plotting package and the OPTISEP package [22,23], which confirmed the expected trends between the objective function (i.e. cost) and the independent variables a minicomputer software [24] routine to perform the optimization of machining conditions in milling has been constructed.

The selection of the optimum machining conditions, including cutter diameter, in peripheral milling is unquestionably a complex task. A typical part which is to be machined using endmills can consist of the following elements :

1) shoulders (see Fig. 2.2.)
2) slots (width of cut is equal to the cutter diameter)
3) pockets

Since the total problem is exceedingly complex, as it consists of the optimization of not only the feedrate, peripheral speed, the choice of cutter diameter, but also of the method of volume subdivision, the "simple" problem of diameter selection was first examined in [21].

2.2.2. Economics of the process.

The costs associated with the milling process may be divided into fixed costs, attributable to non-productive time and raw material costs and variable cost, composed of the following elements : overhead cost of machining, toolcost and tool

changing cost. The variable cost per part would be :

$$C/part = c_h \cdot t_{mach} + c_t \cdot \frac{t_{mach}}{T} + c_h \cdot t_{ct} \cdot \frac{t_{mach}}{T} \qquad (2.8)$$

where C/part : variable cost per part ($/part)
c_h : machining rate ($/min)
t_{mach} : machining time (min)
c_t : cost per tool ($/tool)
t_{ct} : tool changing time (min)
T : cutter toollife (min)

In papers [19] and [20] an interesting plot is depicted (Fig. 2.3.), representing the ratios of permissible feeds per tooth versus cutter immersion (ratio of width of cut to cutter diameter) for constant toollife, constant velocity and constant axial depth of cut. Assuming an unconstrained situation and a small influence of feedrate on toollife (e.g. n=.5), the feed per tooth in full immersion has to be double the feed per tooth in half immersion in order to obtain the same "real" toollife for both situations, so that in the former case four times as much metal has been removed during the toollife. However, a maximum chip thickness constraint (see later) doesn't allow the feed per tooth in full immersion to exceed the maximum feed per tooth in half immersion. Notwithstanding that, the metal removal rate in full immersion, as compared to half immersion, will at least be doubled, partially due to the velocity increase in the slotting situation.

When observing their experimental data, the authors of [18] concluded that "the active cutting life of a tool increases as the width of cut increases. Thus, in general, slotting gives the highest active cutting time, at otherwise identical

cutting conditions". The same observations were made in earlier experiments, as is evident from Fig. 2.4., that the slotting tests give a considerable increase in toothlife. Indeed, the most significant feature of the experiments is the large increase in life which occurs in going from a half immersion test to a full slotting test. This phenomenon has been attributed to the lowering of the thermal stress and strains within the cutter.

For any pocket configuration, as for instance depicted in Fig. 3.3.4., containing a certain depth of cut, the selection of the following parameters needs to be pursued :

1) cutter radius
2) feed per tooth
3) cutting velocity
4) width of cut

It is in the nature of things that the first stroke in a pocketing operation has to be a pass in full slotting. Eventhough this might be a substantial portion of the total pocket, it has not been taken into account during the subsequent derivation. The expression for the cost per volume material removed is :

$$C/volume = \frac{c_h}{v \cdot d \cdot a} + \frac{c_t}{T \cdot v \cdot d \cdot a} + \frac{c_h \cdot t_{ct}}{T \cdot v \cdot d \cdot a} \qquad (2.9)$$

where C/volume : cost per volume material removed ($/in^3)
v : cutter traverse rate (in/min)
a : axial depth of cut (in)
d : radial depth of cut (in)

For a particular cutter radius, depth of cut, cutting velocity and feed, it can easily be deducted from (2.9), that the lowest

cost is obtained by machining with the largest width of cut possible, if toollife were to remain constant. Moreover, slotting tests in [17] revealed a considerable increase in toollife when going from half immersion to full slotting as can be seen from Fig. 2.3. and Fig. 2.4.

While establishing systemmacros that generate the toolpaths to machine pockets, the knowledge of the benefits to be gained while full slotting, has been implemented whenever it was realizable. In case of a constant depth of cut throughout the pocket and bearing in mind the conclusions drawn in the above paragraph, the cost per surface area can be deducted from (2.9).

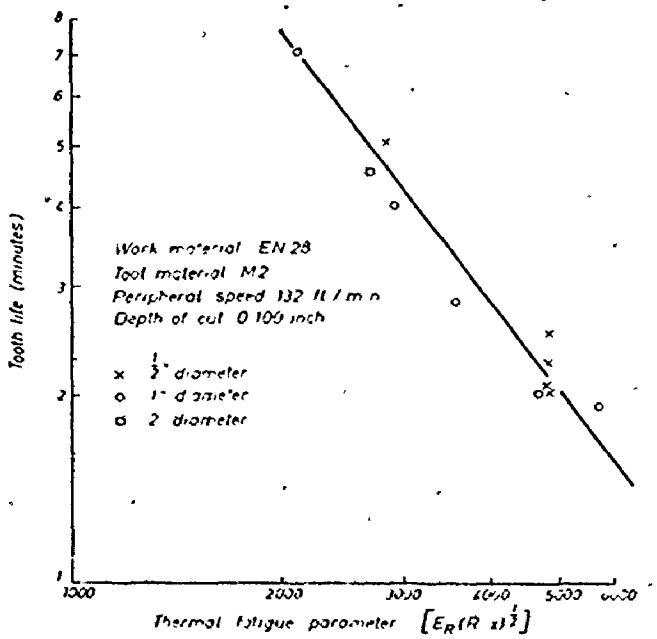
$$\text{Cost/area} = \frac{1}{2R}\left[\frac{c_h}{v} + \frac{c_t}{T \cdot v} + \frac{c_h \cdot t_{ct}}{T \cdot v}\right] \qquad (2.10)$$

where Cost/area : the cost per pocket surface area ($/in^2)
R : cutter radius (in)
T : actual toollife (min)

In comparison with turning, the influence of velocity as compared to feed on toollife will be greater in milling. It is thus evident, from equations (2.7) and (2.10), that any approach to the specification of machining conditions for a certain cutter radius should firstly select the maximum value of feed which satisfies the constraints and secondly select the velocity to give optimal toollife. In [19] and [20], the author, analogue with the turning operation, defines the optimal toollife as :

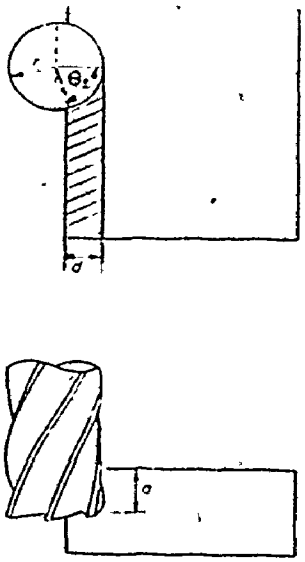
$$T_{(\text{min. cost})} = (z-1)\left[\frac{c_h \cdot t_{ct} + c_t}{c_h}\right] \qquad (2.11)$$

$$\text{with } z \simeq p + \frac{m}{2}$$



Relationship between tooth life and thermal fatigue parameter

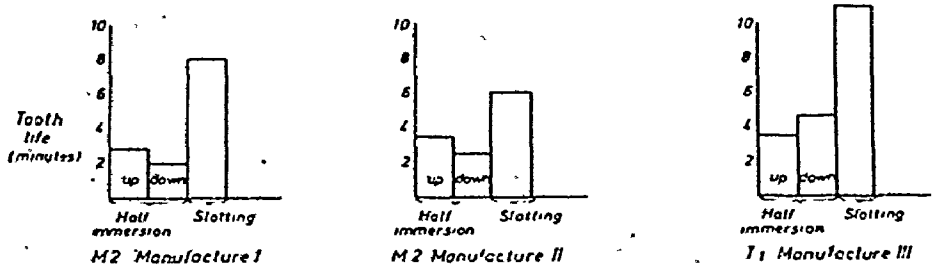
Figure 2.1



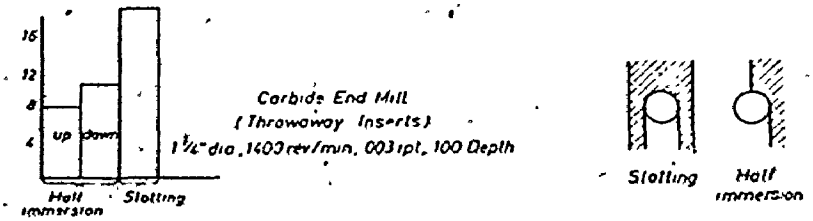
The Basic Geometry of the Peripheral Milling Process

The basic geometry of the peripheral milling process.

Figure 2.2



High Speed Steel End Mills (1" dia 500 rev/min, 004 ipt, 100 Depth)



The Comparison between End Mill Slotting and Half Immersion Tests
(Work Matl EN 28, UTS 57 ton/in²

The comparison between end mill slotting and half immersion tests (work material En 28 UTS57 ton f.in²)

Figure 2.4

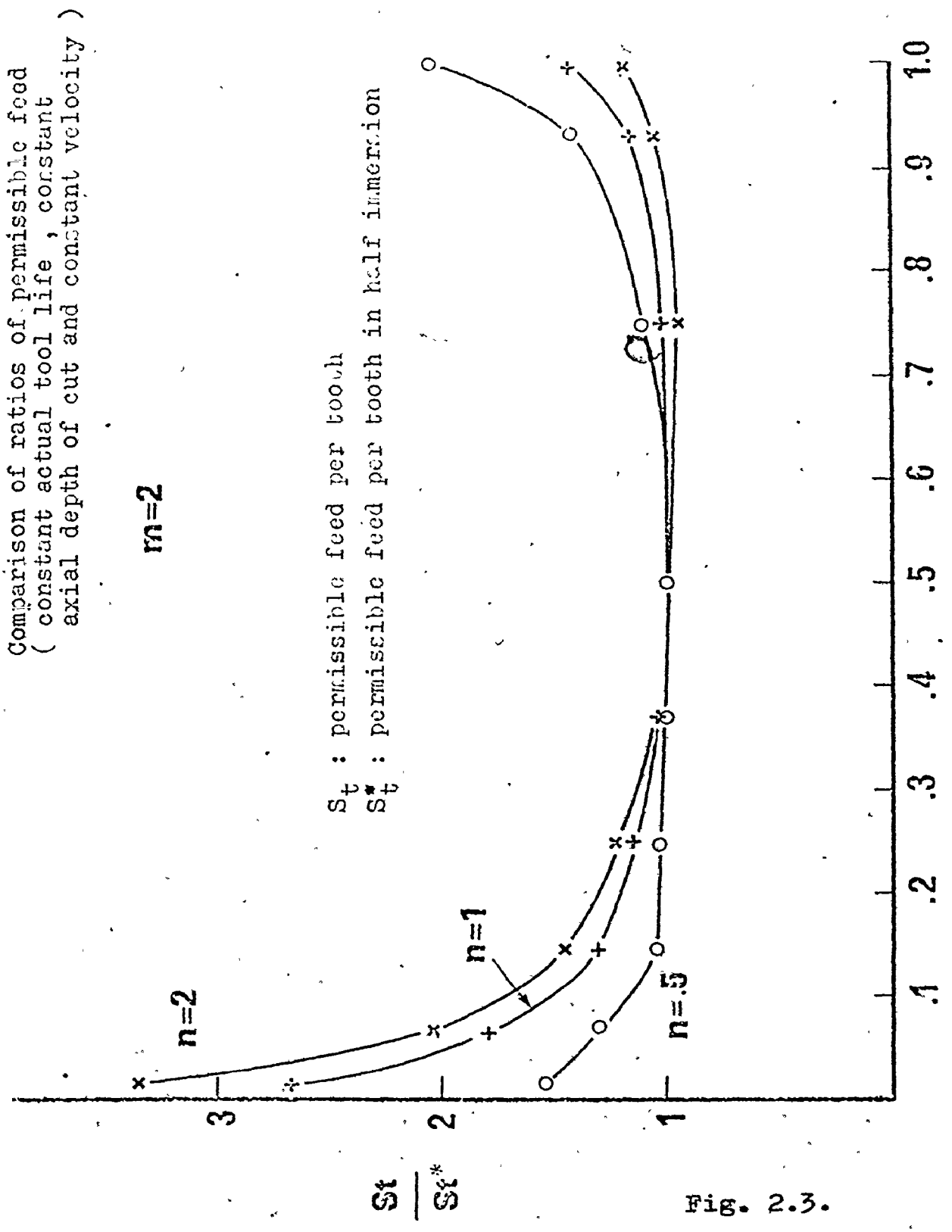


Fig. 2.3.

(copied from reference - works 19,20)

2.2.3. Constraints on the selection of machining conditions.

In practice the usual constraints which must be considered are cutterbreakage (i.e. toothbreakage or shankbreakage), spindle power and spindle torque limitations. An extensive study on the development of the various breakage criteria, aswell as the stipulation of a power and torque constraint, was presented in [19] and [20]. In this work the author will only denote the conclusions of these thorough analysis. The set of practical values for the constraints and limitations, inserted in the optimization routine implemented in the preprocessor, will also be presented.

Toothbreakage might be avoided providing that the maximum undeformed chip thickness (S_{max}) is maintained below a certain limit, i.e.

$$\begin{aligned} S_t \cdot \sin \Phi_s &< S_{max} \quad \text{for } d < R \\ S_t &< S_{max} \quad \text{for } d > R \end{aligned} \tag{2.12}$$

d, radial width of cut
R, cutter radius
S_t, feed per tooth

In the case of the assumption of a brittle failure where the principal stress on the surface must be maintained below a critical value, then the form of the constraint is as follows :

$$M + (M^2 + T^2)^{1/2} \leq \text{const.} \tag{2.13}$$

where M is the resulting moment on the cutter
T is the resulting torque

Assuming a practical case of $\frac{F_{rad}}{F_{tan}} = .3$ (where F_{rad} is the radial component of the cutting force and F_{tan} the tangential component)

the author of [19] and [20] stipulated the ratio of bending moment (M) and torque (T) in the following fashion :

$$\frac{M}{T} = .97 \frac{(1 - a/2)}{R} \qquad (2.14)$$

where l is the flute length (in)
a is the axial depth of cut (in)
R is the cutter radius (in)

As far as the spindle power limitation is concerned, the following equation can easily be derived :

$$P = K \cdot \frac{a \cdot N \cdot n \cdot S_t \cdot d}{3.96 \cdot 10^5} < P_{max} \quad (hp) \qquad (2.15)$$

where N is the number of teeth
n is the rotational speed (rpm)

The maximum torque (T_{max} in lb-in) available can be deducted from the value of the maximum power (P_{max}) available and the critical rotational speed of the spindle drive (V_1)

$$T = K \cdot \frac{a \cdot N \cdot S_t \cdot d}{2} < T_{max} = \frac{P_{max}}{const \cdot V_1} \qquad (2.16)$$

where K is the specific cutting pressure (psi)

The following set of realistic values for constraints has been inserted in the routine. The maximum torque available, the critical rotational velocity of the spindle drive, the maximum principal stress that the cutter can withstand and the maximum chip thickness have been choosen to be respectively 10 hp, 100 rpm, 180,000 psi and .008".

As far as the implementation of this optimization routine in the preprocessor is concerned, the practical aspects of inserting the technological parameters can be found in chapter 3.4. while a detailed build-up of the routine is presented in the flowcharts.

CHAPTER 3

The design of the NC system

3.1. The APT system.

3.1.1. Introduction. [25]

In [25] the author truely states that "APT is as important to part programming as NC is to manufacturing". Indeed, excluding APT from the North American NC scene today would be like taking away the roots from a tree.

The concept of part-description and the problems involved are considerably more important than they seem to be at first. Indeed, it is often remarked that "a drawing is worth a thousand words", however, in many cases it is necessary to convey the image of a drawing in terms of words and symbols.

This latter problem was first brought to light by the advent of NC machines. The APT language was the first approach adopted to the solution of this problem. Amazingly enough, among all NC languages, it is still one of the most versatile and powerful means of part-description. Indeed, the number of variants in the possible ways of defining geometrical entities in APT is extremely large compared to other NC languages and hence the programmer has a great amount of flexibility in the way in which he can describe a part.

To fully demonstrate APT's versatility would lead us outside the scope of this work. However, stating that there are 12, 16 and 10 ways to define respectively a point, a line and a circle, scarcely manifests APT's power. Above all this, the reader has to bear in mind that APT's capability of part-description constitutes only a small fraction of the APT - NC - system.

3.1.2. The building of the APT system. [4,5,8,25,26]

The APT system is made up of two major elements : the general processor and the postprocessor. The general processor, written in Fortran IV, allows an easy interchangability among a large number of computers.

The processing of an APT part program is achieved in four seperate, sequential phases. This four-step procedure is supervised by a control program called "section Ø". Besides controlling the informationflow and the sequence of events among the four sections, the control section (section Ø) assigns the allowable space in the computer storage section.

During the first phase (section I or translation phase), the APT language statements are translated into a numeric form more readily understandable by the phases which follow. This "translation phase", via the part program, establishes the link between human communication to

the computer and NC processor. In this section geometric definitions are reduced to canonical form. Also loops and macro - "calls" (see further) are fully expanded. The translation phase errors are the consequence of part program language and syntax violations. Any error in the translation phase will cause the processor to terminate the job at the conclusion of that phase.

In section II, the arithmetic section, the cutter path, i.e. tool - endpoint - locations and tool axis vectors, is calculated. In this phase the CL data, required to produce the part, are computed by means of the motionstatements, shape of the tool and the specified tolerances (INTOL and OUTTOL statements). During this phase TRANTO° commands and multiple checksurfaces are also evaluated.

In section III, the last section of the general APT processor or also called "editing phase", again a link is established between part programmer and computer. Indeed, the final processor phase edits and prints a listing of the cutter path coordinates (X,Y,Z) generated by section II. This phase (section III), however, can also perform routine data manipulation such as transformation of cutter locations. This modification is able to take place when statements like TRACUT, COPY, etc.... are programmed. Thus the part program-

° TRANTO : Some branching options cannot be executed by the processor before the time the cutter paths are computed. Thus, the statement can only be acted upon on the basis of the results of executing a statement involving toolmovement.

mer automatically obtains an output from this section. If the program does not contain a NOPOST statement, the section III output is automatically fed to the postprocessor (identified by the MACHIN/ statement).

Through the postprocessor (section IV), the APT language provides the part programmer with a means of controlling certain functions at the machine-tool that are auxiliary to the cutter-path relationship. About half of the 300 vocabulary words, that the APT language consists of, is orientated to machinetool functions and transmitted by the APT system to the postprocessor (the other half is used to define part geometry, to specify cutter geometry and cutter-motions, and to command computer system functions). For instance, through proper part program statements, the feedrate is controlled, the spindle rpm (revolutions per minute) can be selected, the coolant can be turned on and off, and, if the machine is so equipped, the tools can be changed. Typical examples of machinetool function statements are

```
FEDRAT/20
SPINDL/1000,CLW
COOLNT/ON
etc...
```

These statements are converted, by the postprocessor, into codes required by the controller. Also, of course section IV "translates" the required motions and functions into the correct format for the particular machinetool. The level of so-

phistication in the postprocessor may vary considerably, however must have the capability of diagnosing simple programming errors, e.g. violation of machine limits, no programmed spindle speed, incorrect tool specification, etc...

Because of the fact that there are capabilities and restrictions peculiar to almost every machinetool/control system combination in use today, not all systems recognize all the postprocessor words available in the APT language. Even different postprocessors may interpret the same word in a variety of ways. For these reasons, the definition of the postprocessor vocabulary words is meaningful only in the environment of a particular postprocessor.

3.2. Advantages of the APT system macros. [28]

Relative to conventional machine-tools, numerical control machine tools are very expensive ($ 10000 for a conventional versus $ 50000 for a 2½ axis numerical control milling machine) but the latter can prove economical if used correctly. Moreover, in the production of complex shapes, digital computer programs must be used to obtain the desired cutter path and to produce the necessary input tape for the numerically controlled machine tool.

In order to be able to write part programs, the part programmer is required to have the knowledge of the machining process, the geometry of the part and a substantial notion of post-processor programs. A complex shape would require a part program of about 1000 lines* to describe the part (geometrical statements) and cutter path (motion statements) in such a way as to produce the necessary output from the computer. It is a fact that in order to produce a complex configuration, up to 50 hours of programming (debugging, plotting and first trial included) may be necessary to produce 1 hour of actual machining-time on the machine tool. The problem of programming numerically controlled machine tools for the production of

* The NC system developed in this thesis would only require a dozen lines of part programming.

complex parts comes down to

a) the computer aided manufacturing language capable of doing the job

b) the personnel capable of programming these complex shapes

c) the high cost of programming

Indeed, it can be estimated that a part programmer costs about 20 $/hour (wages (11 $/hour) + 100% overhead). So one can easily calculate the price of a big program. Needless to say that the high cost of programming can make an operation on a numerically controlled machine tool completely uneconomical.

The author attempted to solve the problem of reducing programming time and, in addition, to enable part programmers, having very little knowledge of digital computer programs, to carry out the necessary calculations. APT contains provision for repetitive programming. Special computer routines are designed to generate whole series of operations from a single command. Four techniques are provided in APT to eliminate tedious, monotonous and repetitive part programming. These routines are PATTERN, LOOPST (and LOOPND), MACRO and COPY. In both macros and loops APT allows the use of conditional branching (IF statement) and unconditional branching (JUMPTO statement).

Analogous to calling a subroutine in a high level computer language, a macro can be called up by a single statement using the major word MACRO and the identifying symbol of the macro. By doing this a group of part programming statements, that form the macro, are called into service. The macro contains variables (dummies) in its call-up statement which can take different values each time the macro is accessed. These variables could be of several types :

a) geometry of the component
b) cutter size
c) angle of rotation of the component
d) translational coordinates of the origin
e) surface finishing allowance
f) postprocessor commands
g) motion commands, e.g. TLRGT (toolright), GODWN (godown)
h) etc...

Moreover, the APT language permits the nesting of macros by recognizing the CALL statement for one macro within the definition of another macro. The maximum level of nesting macros is five. The first macro can call the second one, which can call the third, and so on until a fifth macro has been called. The fifth macro must then return control to the fourth called macro before another macro can be called. The flowcharts (appendix 1) illustrate how the above features have been incorporated in

the program.

The 'APT-macro approach' was investigated and found to be the best approach to the problem, especially when dealing with a system that supports system macros. System macros are macros of which the macro definition is stored in a library so that subsequently executed part programs can directly access them. Macro libraries are mostly stored on an on-line peripheral device of the large-scale main computer. When a CALL statement to a system macro is encountered, the macro is retrieved from storage, defined and executed. The relatively older version of the APT language (APT III) does not support system macros, but in the newer APT IV version one can call up any system macro from the attached macro library.

3.3. Implementation of APT system macros in the NC design.

3.3.1. General description of system macro architecture.

A detailed description of the structure of each system macro can be found in the flowcharts (appendix 1). In the previous chapter (3.2.) the philosophy behind macros has been presented, while in this chapter the basic approach will be introduced.

The division between milling macros and turning macros was not established on purpose, but more due to the fact of a modest start. Indeed, chronologically the relatively simpler turning macros (two dimentional cutting) were developed first. Later on the more sophisticated milling routines were established. Small modifications would allow the latter to be used for complex turning operations (e.g. undercuts).

All macros have been written in APT and are being executed on the CDC 6400 main computer.

3.3.2. Introduction to pocketing (milling) macros. [4,5,29]

The special "POCKET" routine, available in APT III and APT IV, can clean out an area bounded by a maximum of 20 straight lines. The routine is activated by a "POCKET" statement in the part program which defines the points at the vertices of the configuration. APT calculates a cutter center path that will clean out the enclosed area and moves the cutter along that path until pocketing is completed. The calculation of the cutter path is not based on a real spiral, as is the case with the roughing routine SPIRAL, developed in this thesis, but on a decreasing series of concentric polygones. Fig. 3.3.1. shows a pocket with its outermost and next-outermost polygone.

Automatic pocketing as described above can only be done for straight-line configurations. One can bring up that pocketing of configurations that include convex curves can be accomplished by approximating the pocket with straight-line segments by machining an undersized pocket (Fig. 3.3.2.). Not only will the programming of the approximating straight-line segments be laborous and prone to errors because of the fact that the programmer has to calculate the various points on the circle-circumference*, but programming a detailed clean-up can be a real burden.

*At some instances the programmer only defines a circle as being tangent to two lines and having a certain radius, without even knowing the coordinates of the center of the circle. Needless to say that additional calculations have to be executed by the part programmer.

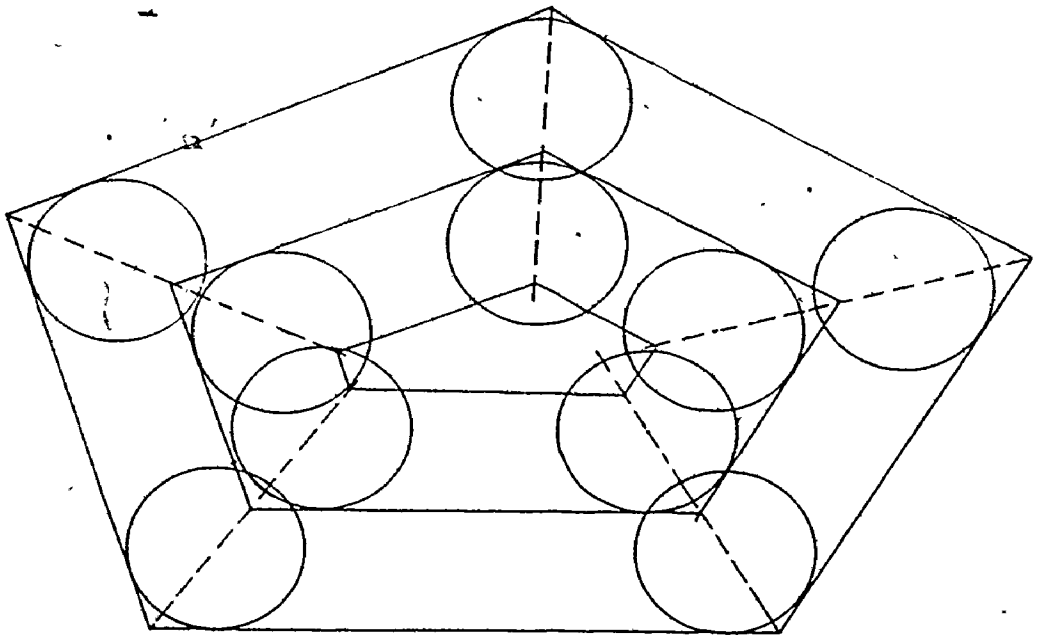


Fig. 3.3.1. The machining of a polygone using the "POCKET" routine.

Notice the overlapping in order to avoid cusps. (more details in section c of 3.1.2.2.)

Fig. 3.3.2. Approximating the pocket with straight-line segments in order to be able to use the "POCKET" routine.

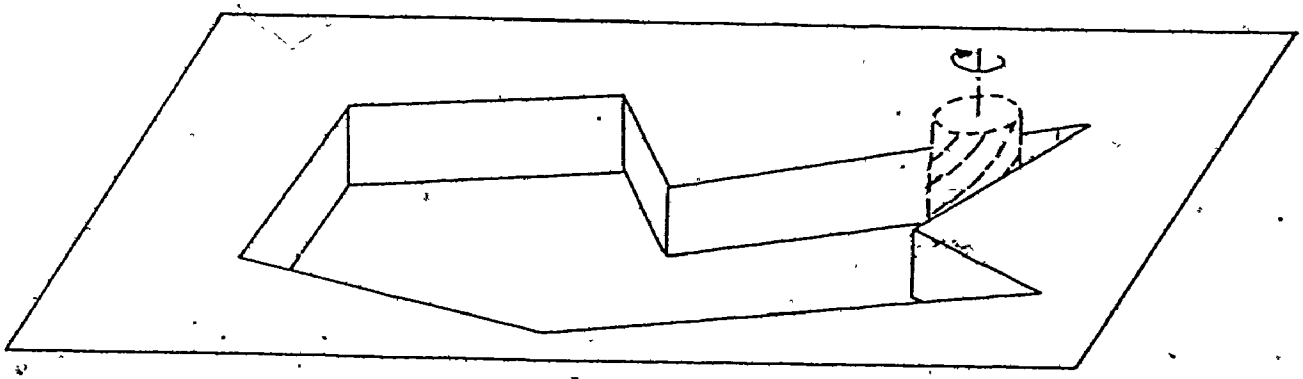


Fig. 3.3.3. No concavity allowed in the geometrical configuration

One should also notice that the approximation can be established using only a limited number of straight-line segments (a total of 20). Moreover, no duplicate points and no more than three points in a line can be used. The most constraining however in this routine is that there can be no concavity in the pocket configuration (see Figure 3.3.3.).

To overcome the above mentioned problems, three sets of macros were designed to produce "volume-elements" rather than tracing toolpaths along "line-elements". This "build-up-blocks" approach results in a reduction in data input in most cases. In order to machine a volume-element, one of the volume clearance routines ought to be called up. The description which follows pertains to the three volume clearance routines developped for pocketing. They are capable of handling two axis contouring in the X-Y plane with incremental straight line motion in Z (2½ axis contouring*). The three pocketing procedures are

1. Unidirectional cutting (RGH1) Fig. 3.3.4.
2. Zigzag cutting (RGH2) Fig. 3.3.5.
3. Spiralling (SPIRAL) Fig. 3.3.6.

RGH1 and RGH2 can handle almost any pocket configuration:

1. There is no constraint on the maximum number of sides on the polygone

*The term "2½ axis contouring" refers to the ability of carrying out normal 2 axis operations while the tool is positioned in a third plane. The third axis of motion is not simultaneous with the other axes (e.g. a screwmotion is not possible.)

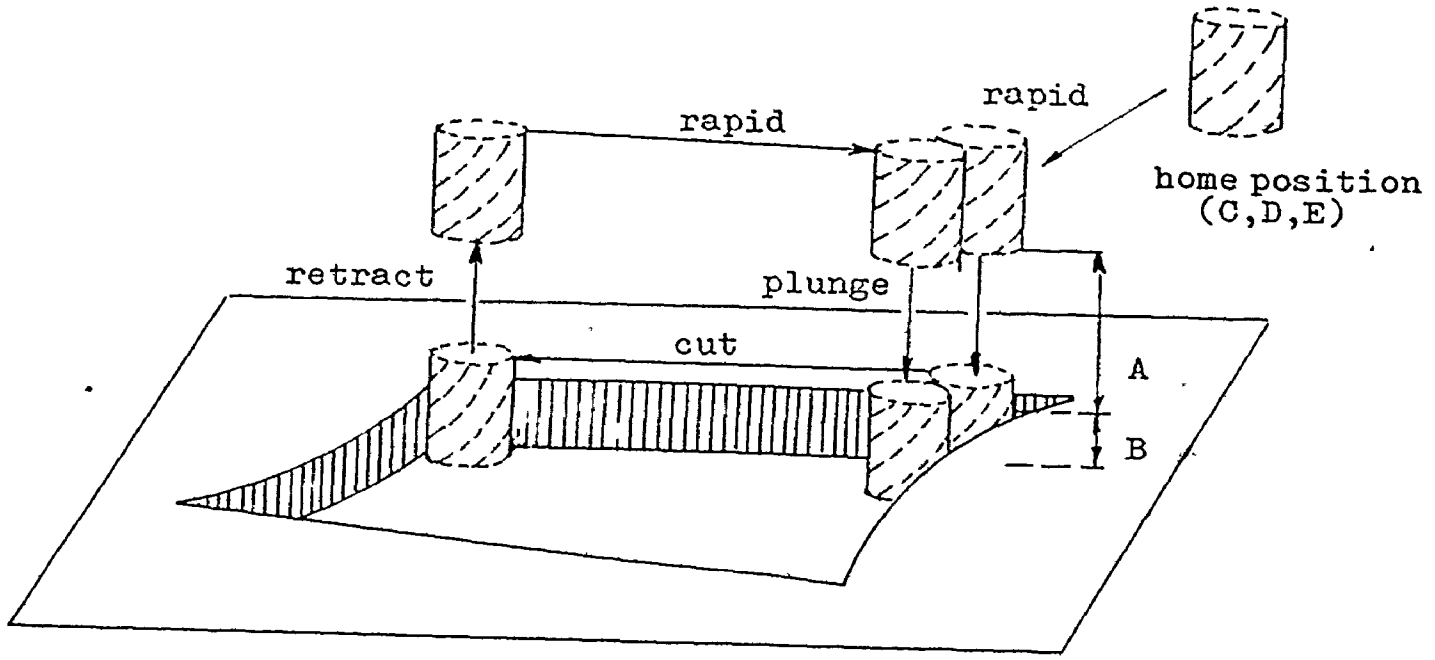


Fig. 3.3.4.a. (isometric)

A : clearance level
B : depth of cut
C : Xstart
D : Ystart
E : Zstart

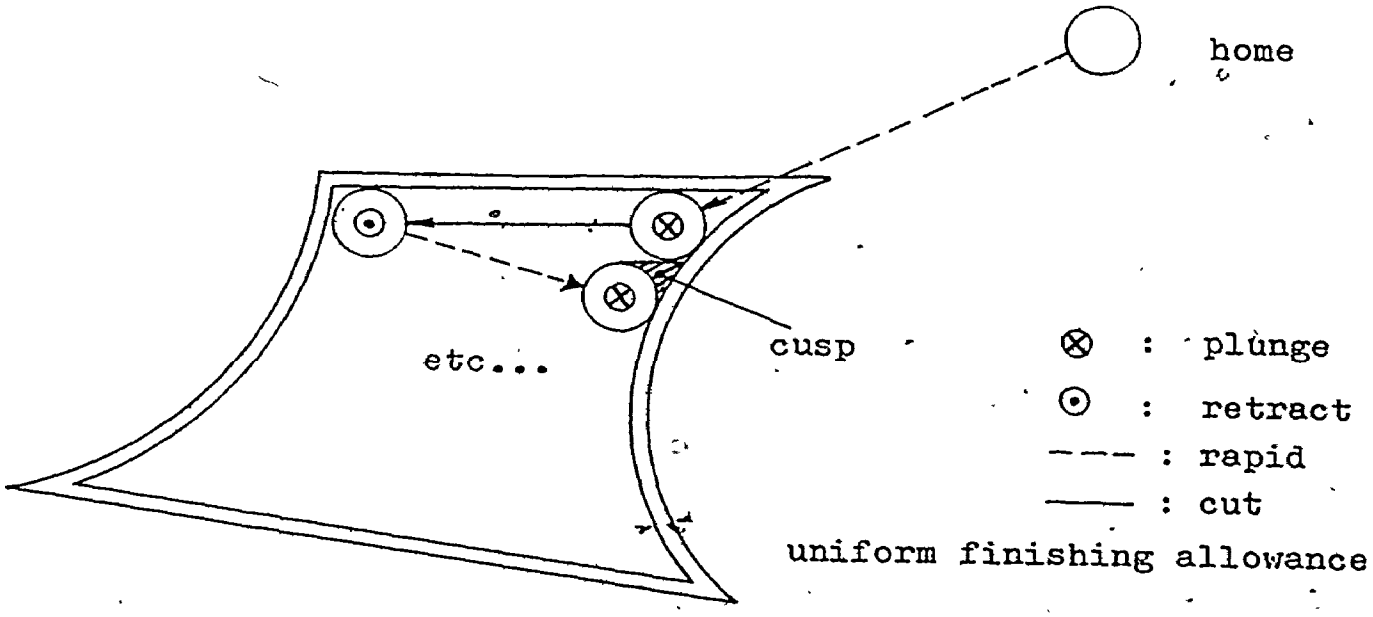


Fig. 3.3.4.b. (topview)

Fig. 3.3.4. Unidirectional pocketing routine RGH1.
(without considering a semi-roughing cut)

2. The boundary of the pocket can consist of straight-line elements as well as circle-arcs.

3. There is no concavity constraint.

SPIRAL is constrainted to a four-side polygone containing two parallel lines but, besides also having advantages 2 and 3 (mentioned above), there is no semiroughing pass needed to clean up the cusps on the boundary (see further), as is the case with RGH1 and RGH2.

In these three pocketing routines, but especially in the SPIRAL set, a modular approach in handling complex pocketshapes is persued. The example depicted in Figure 3.3.7.A. shows how the SPIRAL system would enable the machining of a "three-generation" part ("grandfather" volume, "father" volume and two "son" volumes). Each of them can have a different depth of cut. A composite part (Fig. 3.3.7.A.), although consisting of several independent pockets, can be finished by only calling up FINMAC once. It needs to be stated that the RGH1 and RGH2 (Fig. 3.3.7.B.) sets would have been able to rough out these shapes in one command (CALL/RGH1 or CALL/RGH2), provided that the part programmer contours this 14-boundary-part first, and provided that they have a common partsurface.

The flexibility of calling up volume-clearance routines permits the part programmer to devote more time optimizing the partitioning of the complex shape and to exploit his knowledge and shopexperience more efficiently (e.g. Fig. 3.3.8.). Indeed, the approach of using ba-

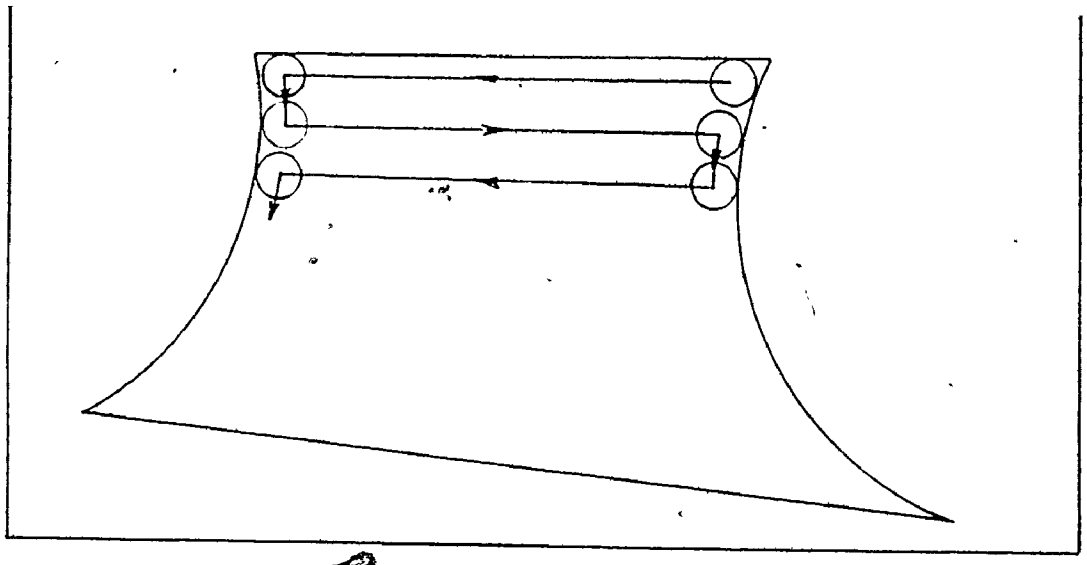


Fig. 3.3.5. Zigzag pocketing routine RGH2

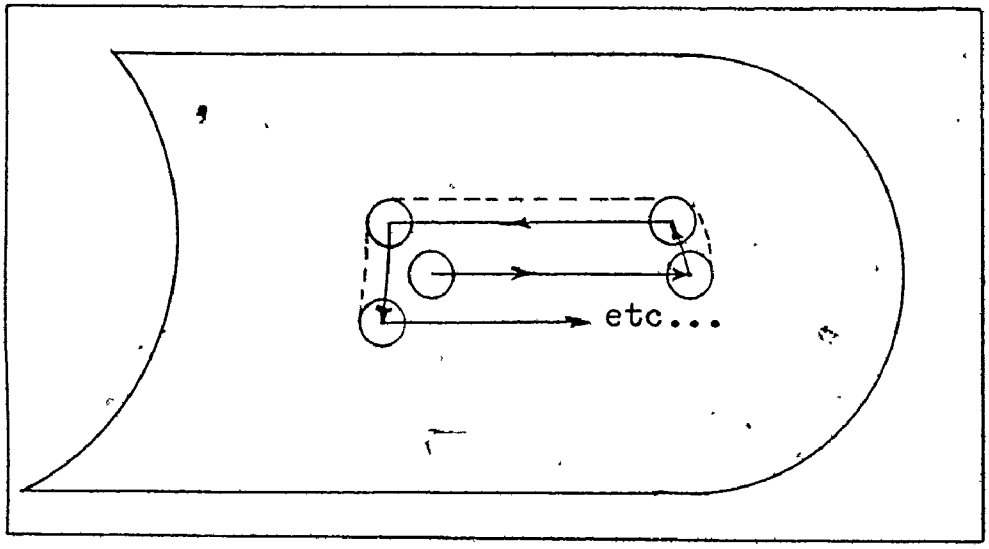


Fig. 3.3.6. Spiralling

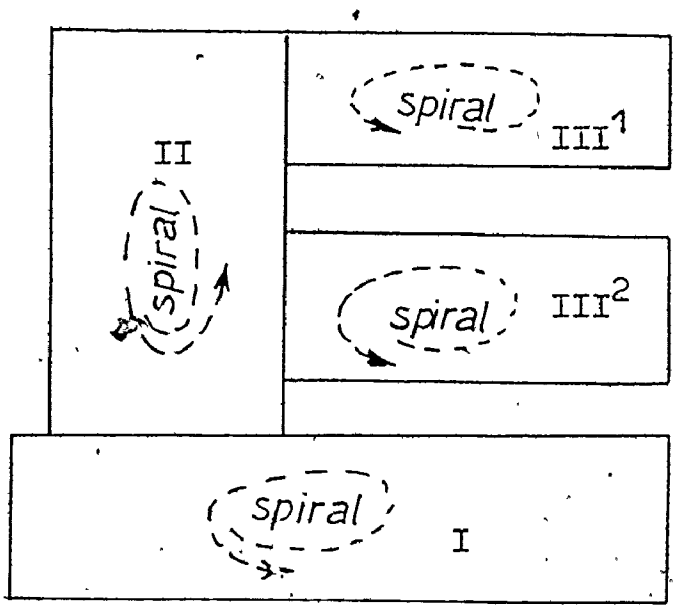


Fig. 3.3.7.A. SPIRAL machining a "3 generation" part.

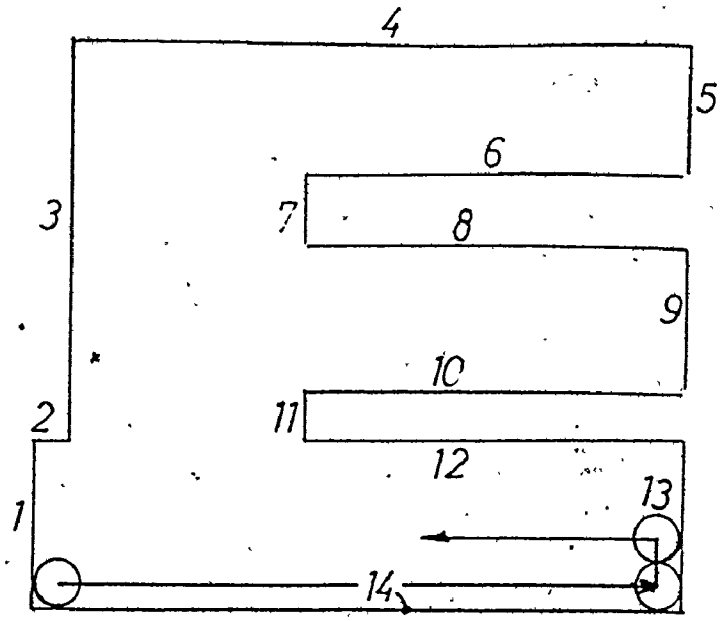


Fig. 3.3.7.B. RGH2 roughing out a 14-boundary pocket

sic geometrical configurations to build up a complex part enables not only greater flexibility, but it also reduces part programming to a minimum and it allows the part programmer to preserve his energy and programmingskills for multiaxes ($\geqslant 3$) part programming.

In the three volume - clearance routines the principal tool cutting direction can be along any line parallel to the X Y plane. This can easily be generalized to any line in space by inserting a TRACUT/ transformationmatrix statement in front of the motionstatements and a TRACUT/ NOMORE after. The ability of transforming the cutterpath points, carried out in the editing phase of the APT processor, just prior to the postprocessing, shows once more the tremendous power of the APT system. Besides translational and rotational transformation, the APT user can obtain mirrorimages of cutterlocations and the section III editing capability even allows him to establish a "nested" copying of his CL data (Cutter Location data).

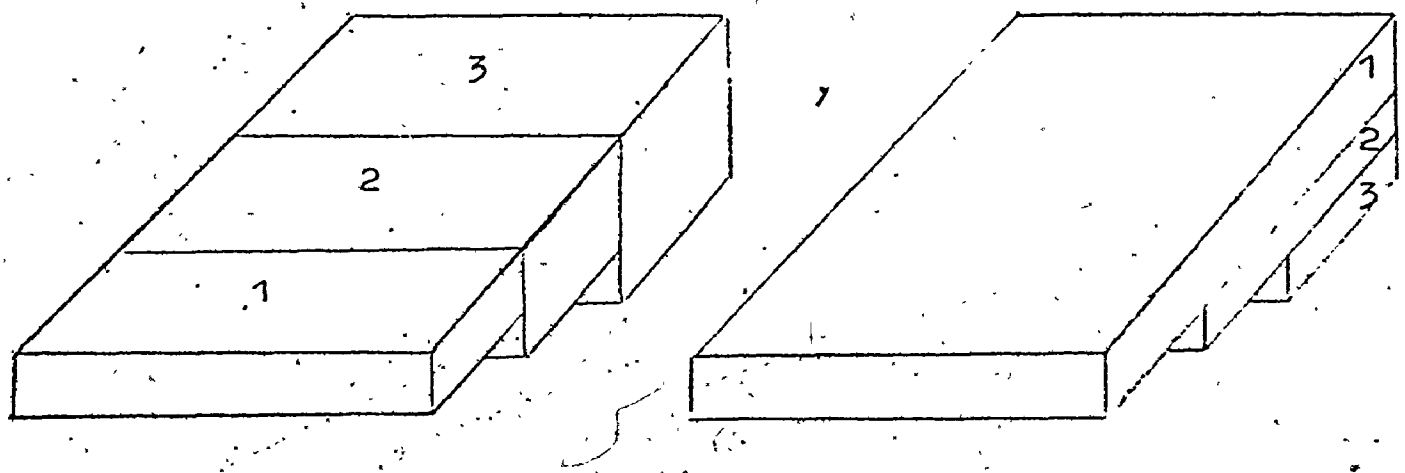


Fig. 3.3.8. Partitioning a complex configuration.

3.3.3. The function of auxiliary macros in part description analysis.

As mentioned in chapter 3.2., it is possible, by creating a library of APT macros, to reduce the time required to program a component to a minimum. Needless to say that there are a number of factors to be considered when writing APT macros that calculate the cutter path of a "general pocket lay-out". The macros must be capable of accomodating any variation of dimensions or positions that is necessary in order to produce all of the configurations required within the limits of the basic geometry (e.g. a pocket (in milling) or e.g. an external shoulder (in turning), etc...). Indeed, these macros must be written in such a way as to cover variations in the number of sides (boundary-surfaces) of the pocket, kind of boundary surface (straight-line segment or circle-arc), depth of cut, cutterradius, spindle speed, feedrate, etc...

In order to succeed in this complicated task, the main macros (RGH1, RGH2 or SPIRAL) call up a common set of auxiliary macros. These supporting macros are MACØ, MAC1, MAC2, MAC4, MAC5 and the submacros (e.g. MACØ1). The lay-out of the supporting macros is sketched briefly in the flowchart of INIT1 (see appendix 1.1.2.).

The main function of the auxiliary macros is to digest the pocket information input (geometry and other) and prepare it in order to accomplish an easy access for the main roughing routines to this information.

Regardless of which volume - clearance routines are used, the geometric description of machined features is always input in the same fashion. The basic geometry of a typical volume - element (handled by RGH1 and RGH2) is shown in Fig. 3.3.10. Besides inputting the geometrical APT definition of every boundary surface, the part programmer needs to describe the pocket because the APT processor only recognizes full circles and straight lines of infinite length (Fig. 3.3.9.). Recently Japanese research engineers have developed an APT like language that recognizes linesegments and circle-arcs.

As will become clear in the examples offered in chapter 4, the user ought to define the boundary surfaces at the beginning of the part program. For instance,

```
CS(1)=LINE/(POINT/0,0,0),ATANGL,135
CS(2)=CIRCLE/CENTER,(POINT/-5,5,0),RADIUS,5
CS(3)=LINE/PARLEL,CS(1),YLARGE,5
  :
 etc...
  :
```

Besides the APT surface definition itself (CS_i), the other basic geometry features of the volume - element are input into arrays (KS, M and N).

The arrayelement KS_i refers to the surface type of the i^{th} surface boundary (CS_i). Meanwhile, these arrayelements procure a concavity code :

$KS_i = 0$ ⟶ CS_i represents a line
$KS_i = -1$ ⟶ CS_i represents a concave circle
$KS_i = +1$ ⟶ CS_i represents a convex circle

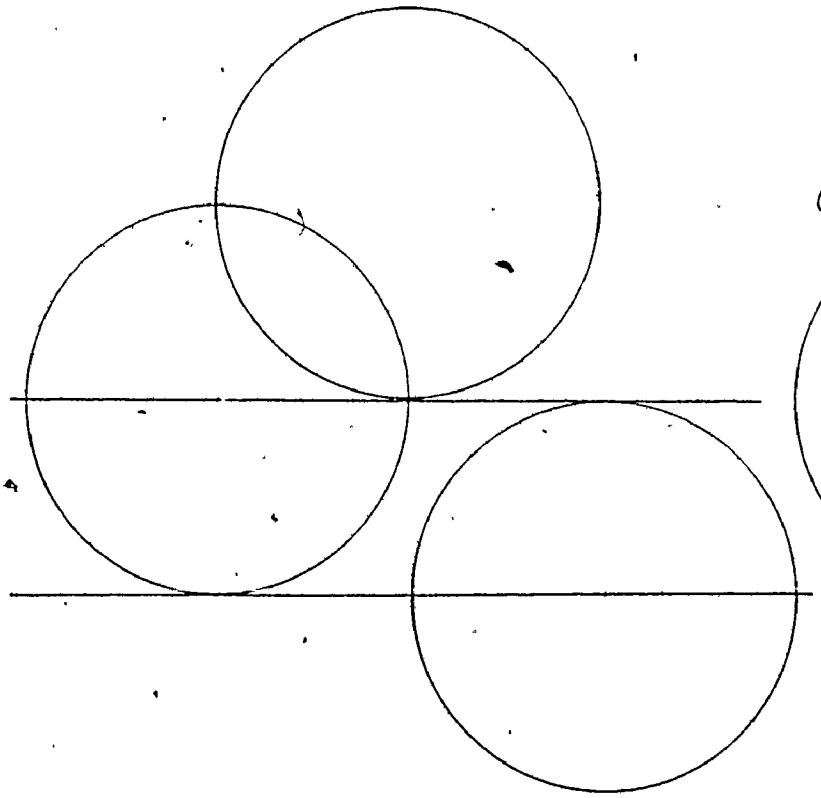


Fig. 3.3.9.

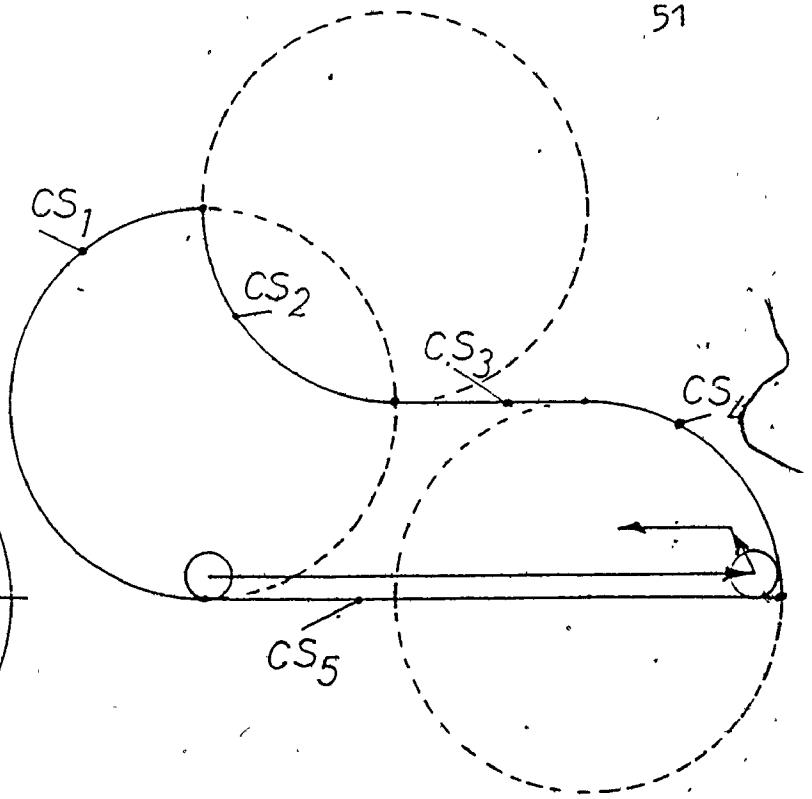


Fig. 3.3.10.

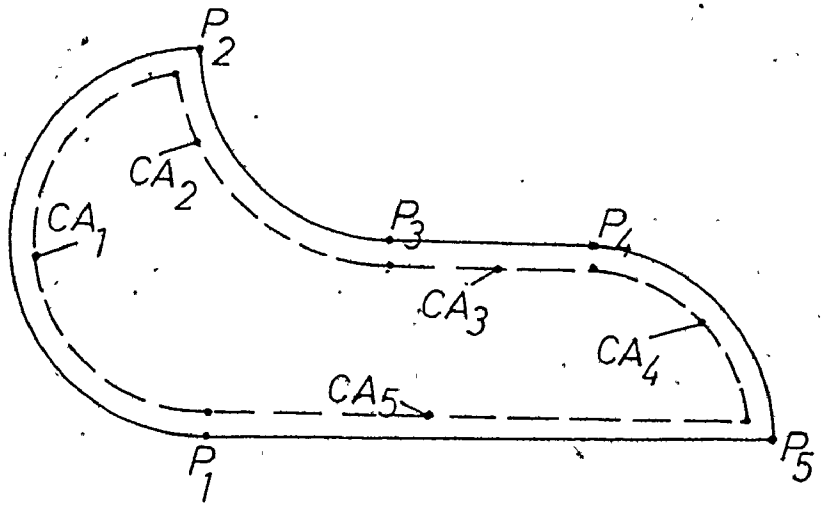


Fig. 3.3.11.

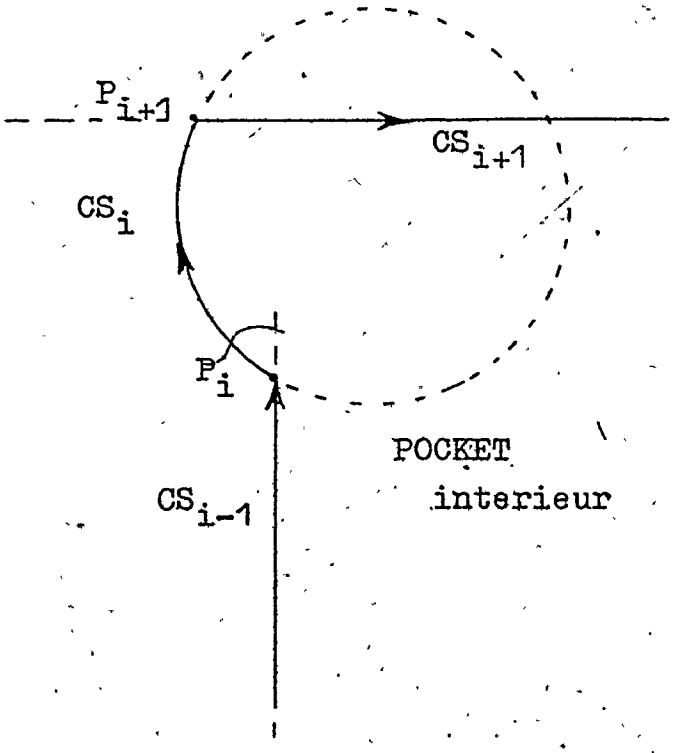


Fig. 3.3.12.
Example to illustrate the
on of N_i

The N_i elements contain information pertaining the intersection points between the adjacent checksurfaces CS_{i-1} and CS_i. Based on the APT vocabulary words XSMALL, XLARGE, YSMALL and YLARGE a code was created to obtain, via the auxiliary macros, the intersection points and meanwhile to establish a unique volume - element. The code works as follows :

XSMALL : $N_i = -2$
XLARGE : $N_i = -1$
YSMALL : $N_i = +1$
YLARGE : $N_i = +2$

and is illustrated by Fig. 3.3.12. where N_i=+1 and N_{i+1}=-1.

The M_i elements contain information pertaining the directional modifiers, which select the proper path of travel, in every APT motion statement. In order to achieve the stipulation of a semiroughing or a finishing pass, the knowledge of those parameters is indispensable. The key to this code is :

GOBCK : $M_i = -2$
GOFWD : $M_i = -1$
GOLFT : $M_i = +1$
GORGT : $M_i = +2$

It is rather obvious that the element M_i and M_{i+1} in Figure 3.3.12. carry the values +1 and +2 respectively.

The auxiliary macros will calculate the vertices of the pocket (P_i), the offset surfaces (CA_i) aswell as other necessary information for the clearance macros. Besides determining the vertices P_i and the offset surfaces CA_i (see

Fig. 3.3.11.), the auxiliary macros (MACØ, MAC1,...) also calculate the limits of the cuttercenter-coordinates at the pocket vertices. Fig. 3.3.15 shows an example of the stipulation of these coordinate-points. It can easily be noticed that for vertex-angles $\leqslant 180°$, the limit coordinates are identical ($X1_i = X2_i$ and $Y1_i = Y2_i$). Fig. 3.3.11. gives an example of how the auxiliary macros have digested all input and defined the volume element as a unique configuration.

Practical realisation:

In our case, the APT part program was compiled and executed by an APT processor and postprocessor on-line, via magnetic tape, at the McMaster University CDC 6400 large scale computer. However, as will become clear in chapter 3.4., a part program can be established (and preprocessed) on a smaller digital computer.

Upon investigation a minicomputer is found to be very successful in supporting part program preparation.[30] Provided that the part programmer has access to a general purpose digital minicomputer, computational algorithms and software routines can be developed to relieve the part programmer of the burden of inputting small-detailed pocket-information into the part program. While letting the part programmer, in an interactive way, contour the pocket, the minicomputer software achieves all the necessary part program information by manipulating the string variables, in-

put by the user. This represents in fact only one of the minor advantages of computer-aided part program preparation. An analysis of the above mentioned minicomputer software, developed by the author, as also a detailed study of part program preparation by means of a supporting minicomputer in a conversational mode, will be dealt with in chapter 3.4. Provided that the part programmer has access to such a minicomputer, the preparatory information to the volume-clearance routines (CS_i, M_i, N_i and KS_i) is inserted and edited automatically.

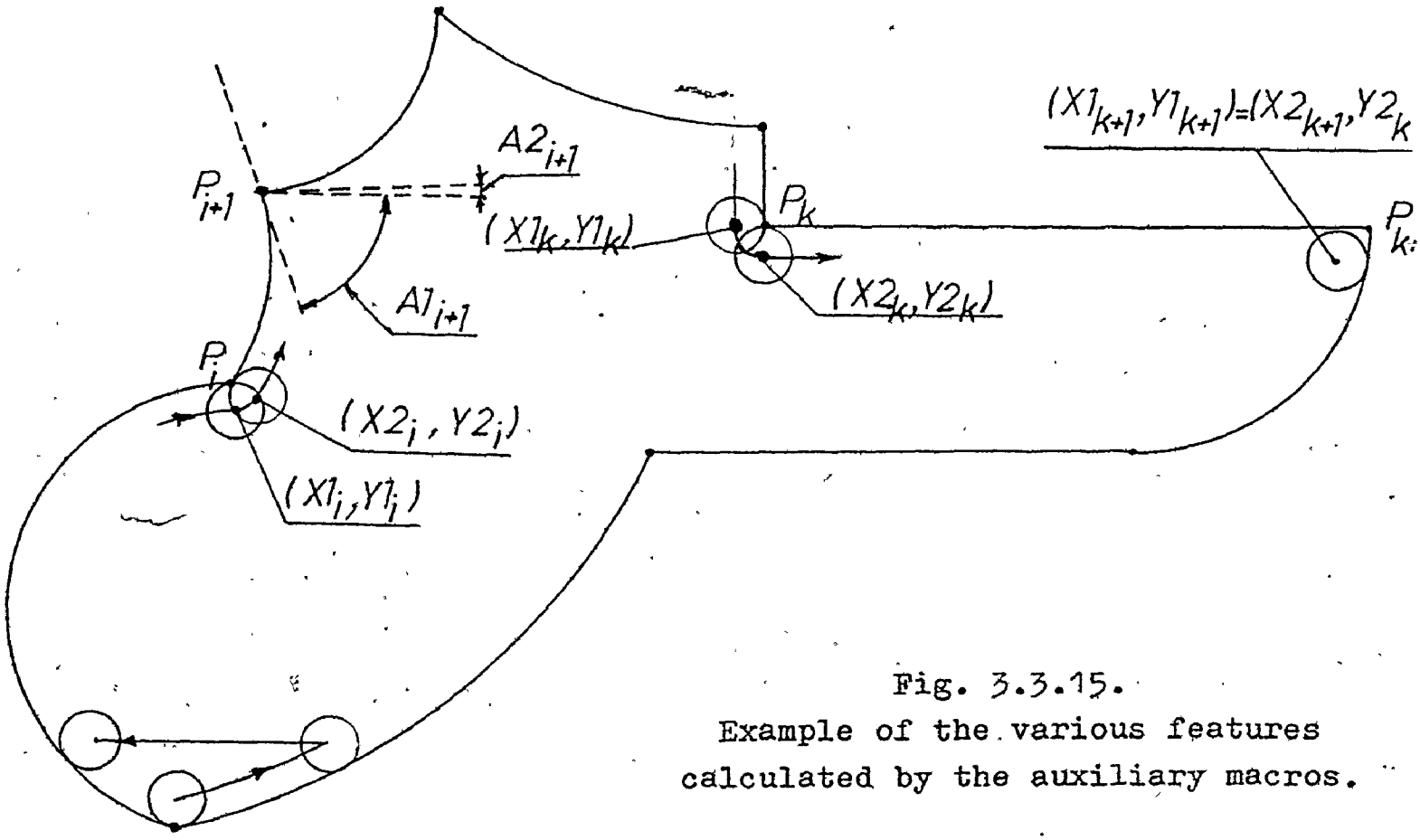


Fig. 3.3.15.
Example of the various features calculated by the auxiliary macros.

It can be noticed that for inner pocketangles smaller or equal to 180°, the "limit-coordinates" are equal : $(X1_i, Y1_i) = (X2_i, Y2_i)$

3.3.4. Architecture of the pocketing macros. [38]

A well-outlined description of the volume-clearance milling macros can be found in the flowcharts of appendix 1.

In establishing the general pocketing macros, including FINMAC, the "inlayed drive surface" was found to be the safest technique when generating toolpath statements. Consider the toolmovement in Fig. 3.3.13.A. [38]. In order to continue along L2, the cutter does not have to go all the way to P2. An auxiliary drive surface can be inlayed between L1 and L2 (Fig. 3.3.13.B.). Fig. 3.3.14. shows an exerpt from the semiroughing macro FINMAC. Provided the cutterradius is small (R1), the motionstatements presented in Fig. 3.3.14.A. will generate a correct toolmotion. A larger cutter (R2), however, will make the processor hang up and create an APT section II error message. Fig. 3.3.14.B. represents the technique implied in this thesis.

In contrast with Fig. 3.3.14.A., one can notice in Fig. 3.3.14.B. that the cuttermotion is directly defined by the cuttercenter (TLON : tool on drivesurface, instead of TLRGT : tool right tangent to drivesurface during movement). This prevents the cutter, at the end of the cuttermotion, of not being in tolerance of the subsequent drive surface. This burden was the inspiration for the use of the CA_i surfaces (see above).

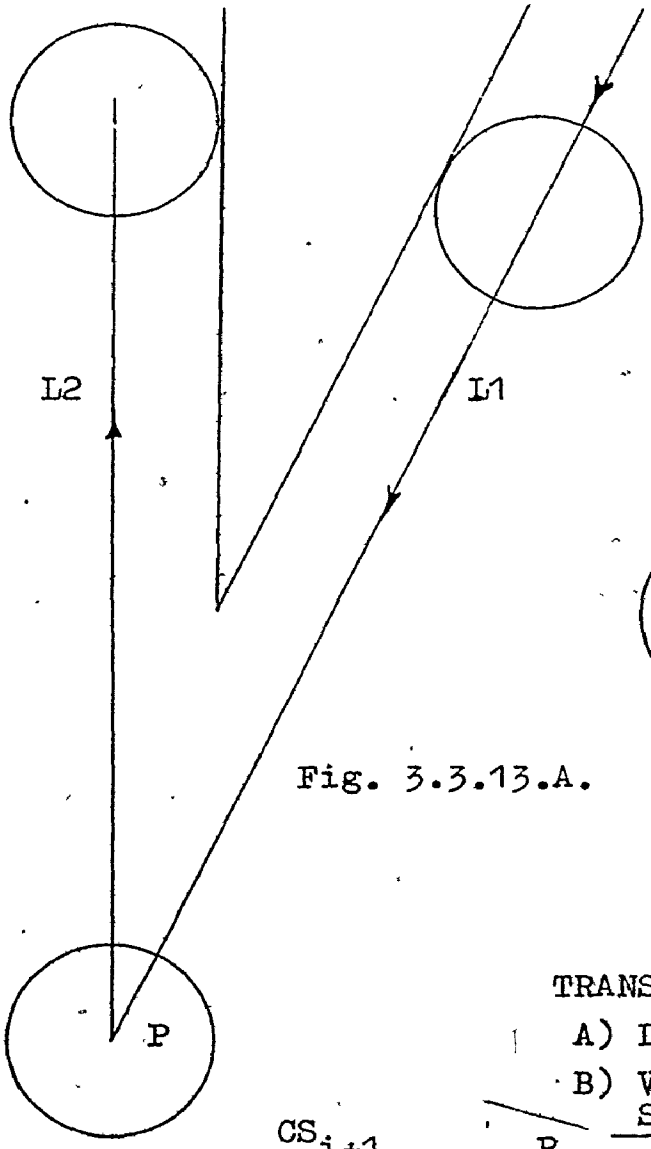


Fig. 3.3.13.A.

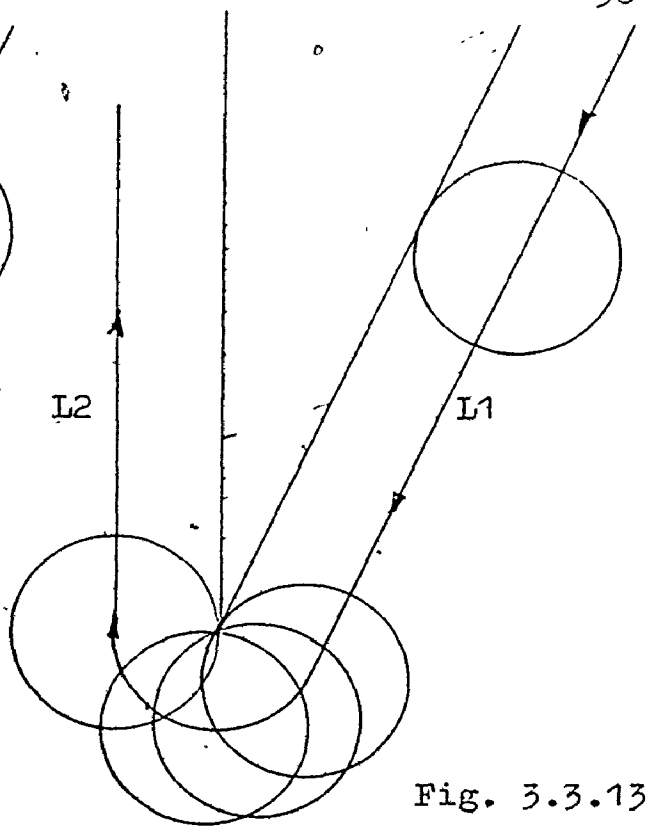
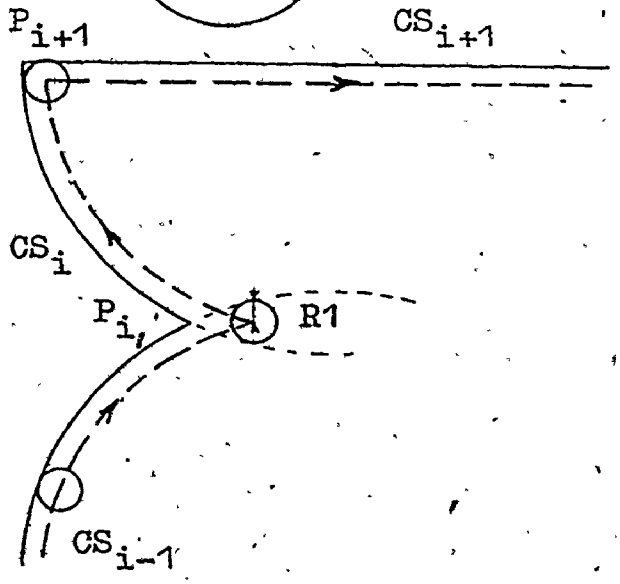


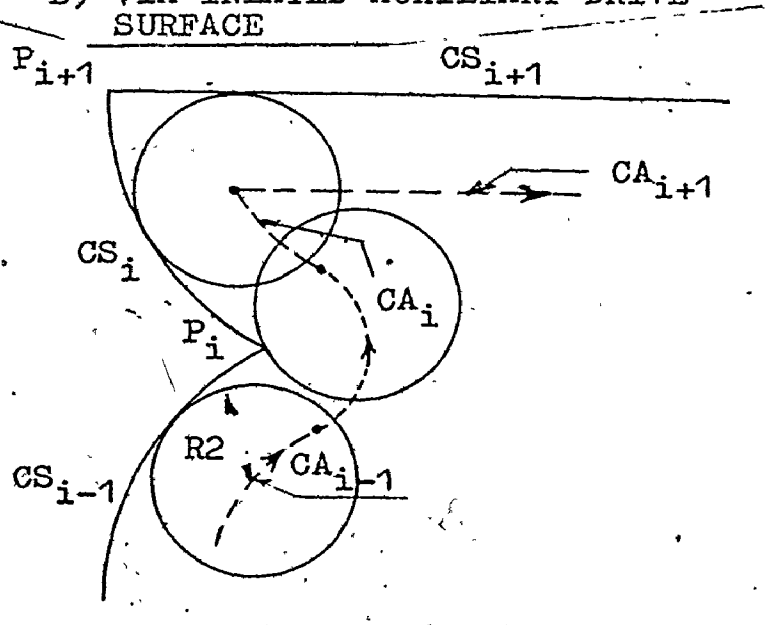
Fig. 3.3.13.B.

Fig. 3.3.13. :
TRANSFER OF TOOL FROM L1 TO L2
A) DIRECT
B) VIA INLAYED AUXILIARY DRIVE SURFACE



TLRGT,GORGT/CS(I-1),PAST,CS(I)
TLRGT,GOLFT/CS(I),TO,CS(I+1)
NOT VALID FOR LARGE CUTTER !

Fig. 3.3.14.A.



TLON,GORGT/CA(I-1),ON,
(CIRCLE/CENTER,P(I),RADIUS,R)
TLON,GOFWD/(CIRCLE/CENTER,
P(I),RADIUS,R),ON,CA(I)
TLON,GOFWD/CA(I),ON,CA(I+1)

Fig. 3.3.14.B.

The reader should notice that the need for implementing the two phenomena mentioned above (inlayed drive surface and CA_i surfaces) is due to the fact that circle-arcs are allowed as boundary surfaces. Needless to say that the primitive built-in APT POCKET routine does not have to deal with these inconveniences.

3.3.4.1. The semiroughing macro : FINMAC.

Cusps remain at the boundary surfaces after having roughed out a pocket by one of the roughing routines RGH1 or RGH2 (Fig. 3.3.4.). In order to get rid of these uncut portions, a semiroughing cut ought to be held either before or after the roughing cuts. A preceding semiroughing cut seems more favorable because it doesn't show the intermittent cutting features of a subsequent semiroughing cut.

It needs to be stated that the semiroughing macro FINMAC is capable of executing a finishing cut. Moreover, a composite part (Fig. 3.3.16.), although consisting of several independent volume-elements, can be finished by only one call to FINMAC.

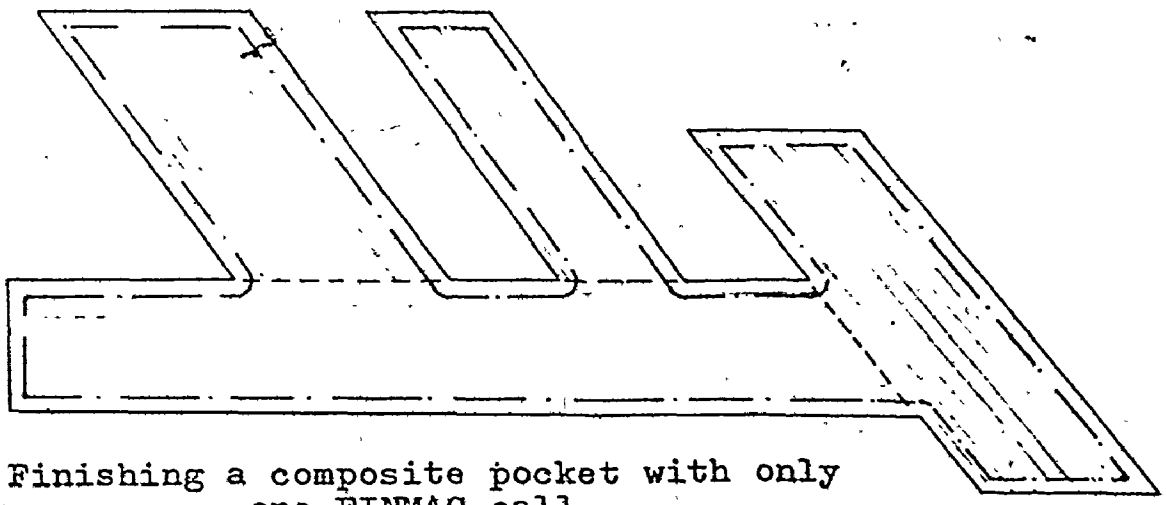


Fig. 3.3.16.

Finishing a composite pocket with only one FINMAC call.

3.3.4.2. The philosophy behind and the mode of usage of RGH1.

As mentioned above the principal tool cutting direction for the three volume-clearance routines (RGH1, RGH2 and SPIRAL) is along the X-axis. Because of the versatility of APT, a transformation of coordinates can make any line in space be parallel to the X-axis.

The RGH1 roughing macro enables the machining of a pocket while the cutter performs a sequential cutting motion from the lefthandside (Xsmall) to the righthandside (Xlarge) of the pocket (i.e. always in the direction of the positive X-axis). In between the metal removal motions, the tool is retracted to a height A (macro variable) above the surface and shifted to (rapid horizontal movement) a point vertically above its new starting position. After plunging to the partsurface of the pocket, the cutting cycle repeats itself (see Fig. 3.3.4.A.).

In order to ensure full slotting (axial width of cut is twice the cutterradius), the subsequent cutterpaths had to be parallel and twice the cutterradius apart. Despite APT's limited abilities in performing some algebraic manipulations, the total number of laps (K3) to cover the whole pocketarea is calculated first (Fig. 3.3.17.). Subsequently the starting points (U_i) and end points (V_i) of the various laps are determined (Fig. 3.3.17.). After having stipulated the necessary parameters, the actual cuttermotions are executed. In comparison with the software logic implemented in RGH2, the RGH1 program built-up is far less sophisticated.

Eventhough the milling roughing macros RGH1 and RGH2 can handle rather complex pocket shapes, partially due to the section III TRACUT facility, they are not almighty. In the local coordinate system of each pocket, the highest and the lowest (Ylarge and Ysmall) point have to be intersection points. When contouring clockwise, the Y-value of points on the boundary surface must steadily increase or stagnate until the highest intersection point is reached and from then on decrease (or stagnate) until the lowest intersection point (starting point) is encountered again.

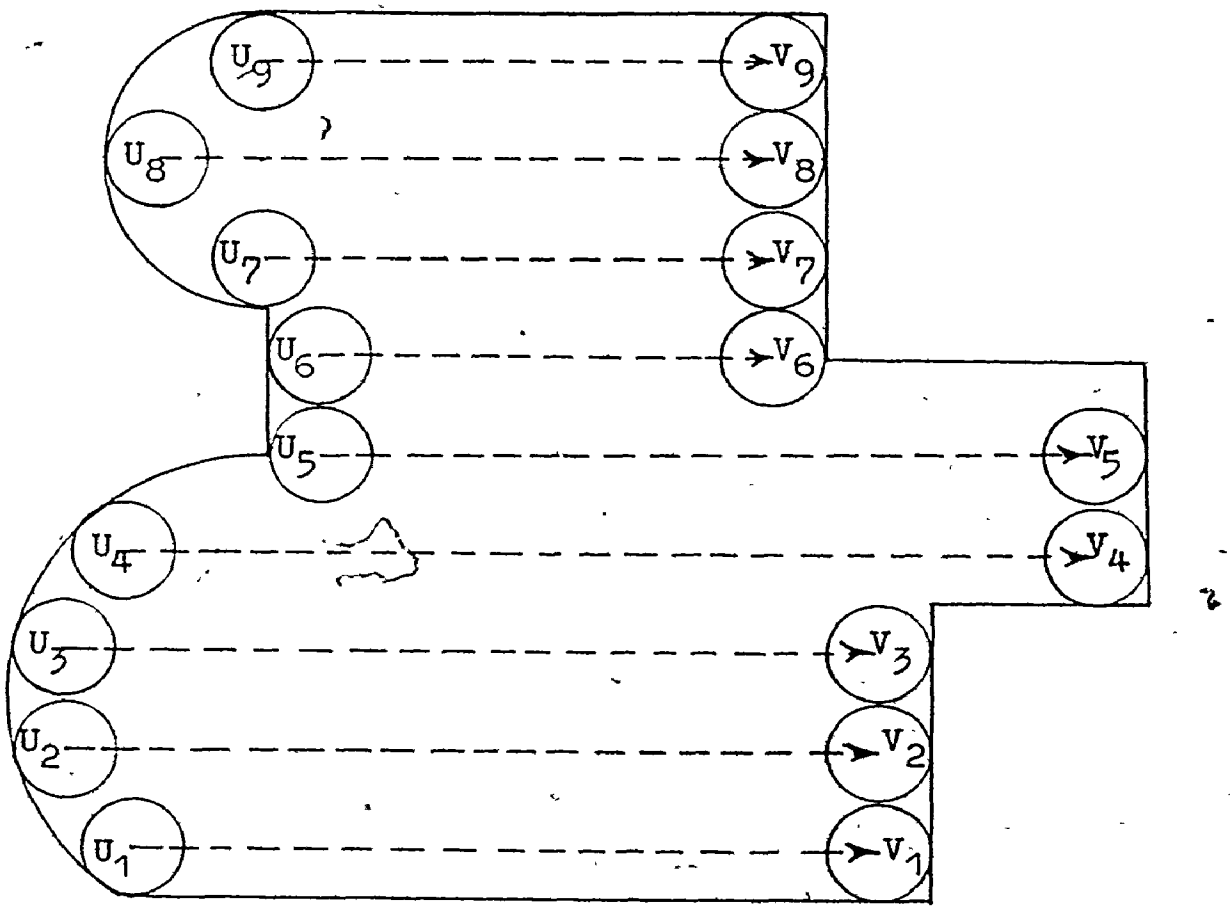


Fig. 3.3.17. Example of the stipulation of the number of cutterpaths (K3=9) and their starting point (U_i) and endpoint (V_i).

3.3.4.3. The philosophy behind and mode of usage of RGH2.

The zigzag technique is often applied in pocketing. The software logic involved is far more complicated than the one implemented in RGH1. RGH2 is especially beneficial for two reasons :

- this volume - clearance facility machines a pocket faster than any other method (for the same feedrate)
- it is also the method in which the tool almost always cuts under quasi - optimal conditions.

When making a comparison between the four different milling volume - clearance routines, the superiority of RGH2 results in the shortest machining time. Indeed, compared to RGH1, the tool only plunges once and isn't released from the partsurface until after the job is terminated. In contrast with SPIRAL, RGH2 always machines in full slotting (i.e. optimal cutting conditions; see chapter 2). It needs to be added, however, that, as is the case with RGH1, a semiroughing pass precedes the actual zigzag motion. Indeed, as explained in a previous paragraph (3.3.2.), the phenomenon of cusps occuring along the sides of the cutterpath has been avoided by executing a preceding cut (see also Fig. 3.3.15.).

A powerful feature of this technique (also practicable by RGH1, not by SPIRAL though), is that this algorithm can deal with islands economically (see Fig. 3.3.18.). An extended description of this technique can be found in flowchart 1.1.3.3.

3.3.4.4. The philosophy behind and mode of usage of SPIRAL.

The SPIRAL method is a typical offsetting technique. It contains approximately the same philosophy as the APT POCKET routine. The latter pocketing algorithm doesn't really produce a spiraling cutterpath. It is constituted of a series of concentric polygones (see Fig. 3.3.1.).

Initially, our goal was to generate a general pocketing routine (containing circle-arcs aswell as straight lines for boundary surfaces) in which the cutter would remove material while performing a spiral-like cutter path. Very soon we learnt that the generality of the problem gave rise to tremendous complex geometrical calculations. To constraint our technique to a straight-line polygone (nevertheless including concavities) would bring its implementation very close to the already existing POCKET routine, and was rejected for this reason.

Finally our preference went to a modular built-up approach, consisting of elements (pockets) containing only 4 boundary surfaces of which 2 are parallel. It is in the nature of pocketing by this method, that uncut portions may remain if careful consideration is not taken to see that succesive cutter paths are not close enough to each other. As depicted in Fig. 3.3.19. , the reader will notice that full slotting does not guarantee that there will be no cusps left (except for the case in Fig. 3.3.20.). To safeguard against leaving material uncut in the figure a cutter-offset has been programmed into the SPIRAL routine. This feature checks for

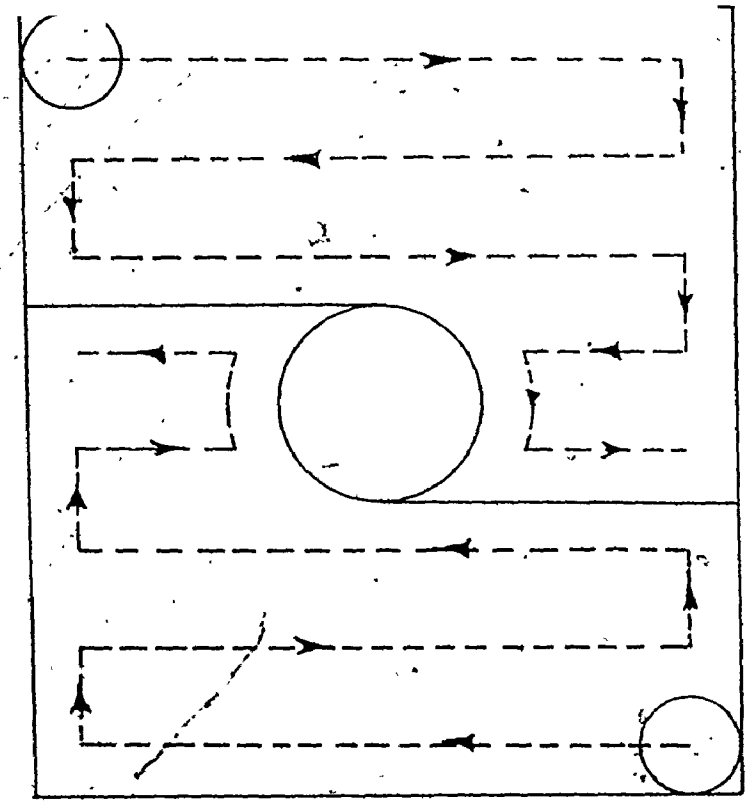


Fig. 3.3.19.
Material left in pocket

Fig. 3.3.18.
Routine RGH2 can handle islands in pockets by considering the configuration as existing out of 2 pockets.

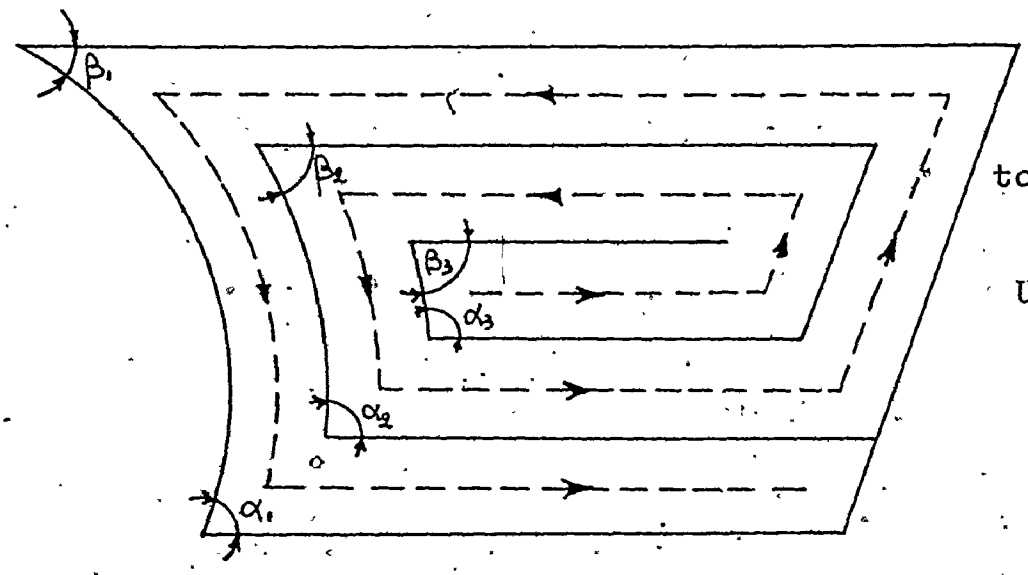


Fig. 3.3.21.

$\alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 < \alpha_5$

α_1 is the smallest angle

$\beta_1 < \beta_2 < \beta_3 < \beta_4 < \beta_5$

β_1 is the smallest angle

the smallest angle in the figure and reduces the cutter-offset accordingly. Indeed, the offset of subsequent concentric cutterpaths is determined by the smallest angle of the quadrilateral configuration.

If surfaces 1 and 3 in the four-sided configuration are concave circle-arcs or straight-line segments, the smallest angle in the quadrilateral figure is located on the outer polygone (Fig. 3.3.21.). The calculation of the smallest angle encountered along the cutterpath becomes extremely complicated when dealing with convex circle-arcs as boundary surfaces. As depicted in Fig. 3.3.22., one notices that the inner angles of the concentric polygones decrease when shifting towards the inside of the figure. Once the smallest angle defined, the minimum cutter-offset can be calculated by the following formula :

$$(\Delta R)_{lim} = R * (1 - \sin \frac{\alpha}{2})$$

$(\Delta R)_{lim}$: minimum cutter-offset in order not to leave any cusps along the cutterpath, (in)

R : cutter radius, (in)

α : smallest angle along the cutterpath

A detailed calculation of the offset (ΔR) and all the auxiliary geometrical manipulations can be found in appendix 3.

Having calculated the width of cut, a grid can be constructed on which the cuttercentre will travel while roughing out the pocket. The pattern will be slightly dependent on whether or not there's an even or uneven number of passes (quasi-circles) involved. Fig. 3.3.21. and Fig. 3.3.22. depict both

cases. An extended analysis of the SPIRAL set of APT macros is available in the flowcharts.

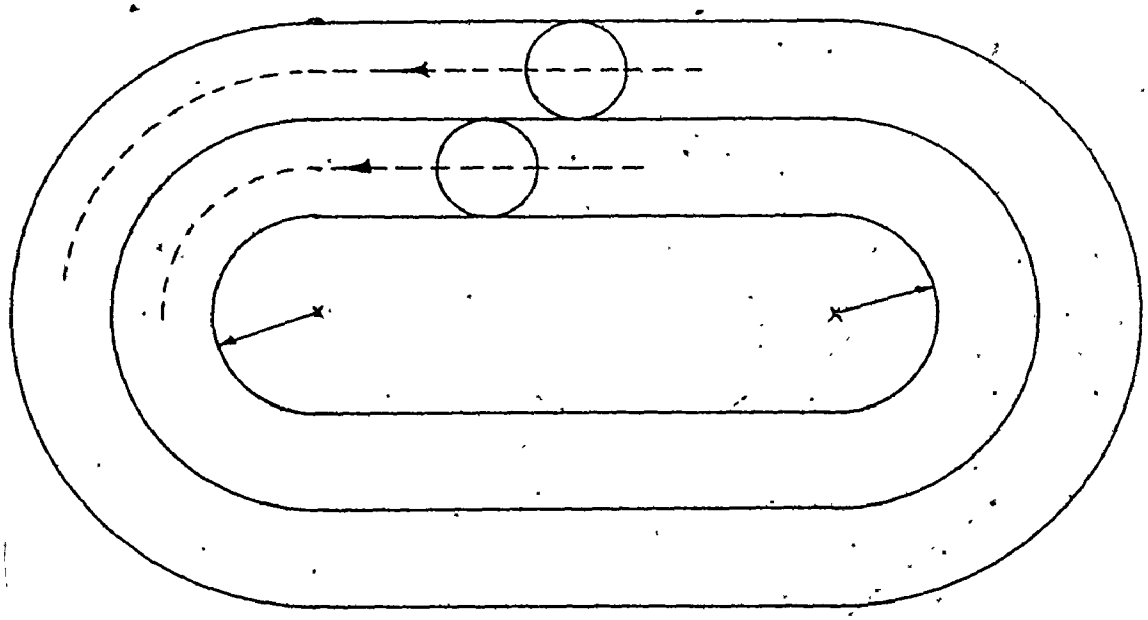


Fig. 3.3.20.

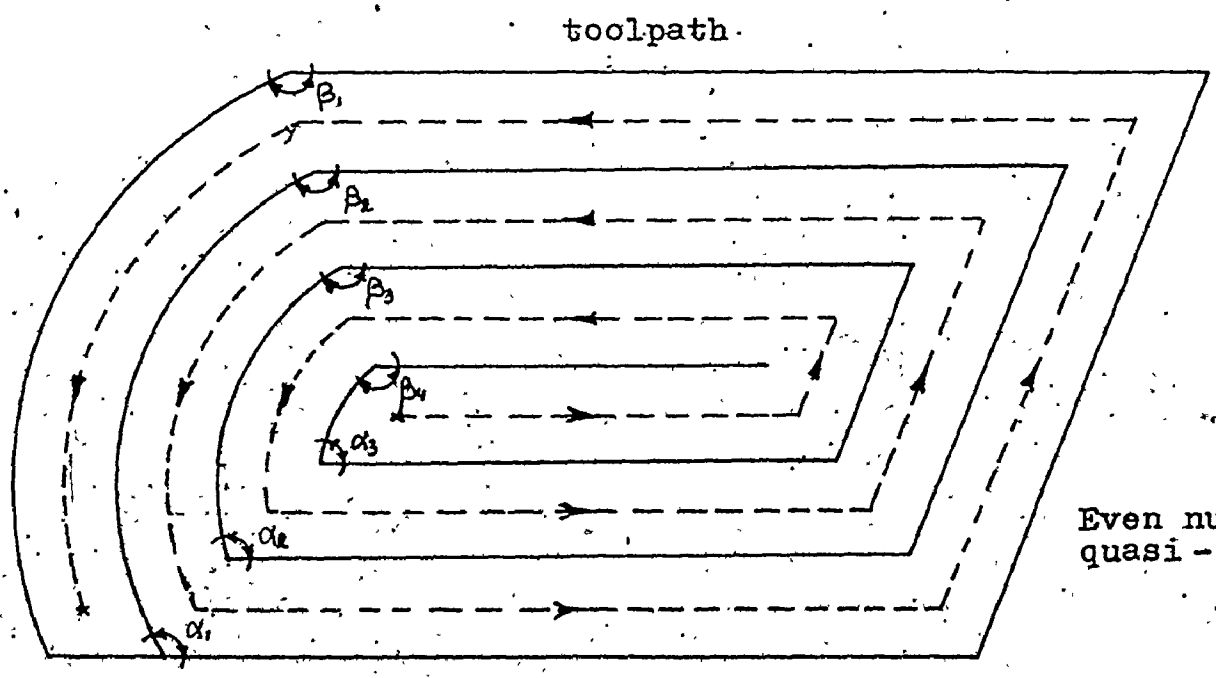


Fig. 3.3.22.

$\alpha_1 > \alpha_2 > \alpha_3$

$\beta_1 > \beta_2 > \beta_3 > \beta_4$

3.3.5. Analysis and architecture of the turning macros. [7]

Unlike other NC systems (e.g. EXAPT), the APT processor does not make the distinction between milling and turning operations. The post-processor (section IV) digests the CL data (output section III) and is capable of producing the NC tape.

In contrast with three dimentional configurations, the description of pure rotational components concerns only two coordinates, hence the APT language is based on definitions of points, lines and circles, while the part surface always remains the Z = 0 plane (in the part program coordinate system).

In order to produce a turned part with the existing APT system, the relevant features of the component need to be defined and the user then has to program toolmotion statements to guide the tool in the required path. This is a rather simple task for a finishing cut, however, programming a roughing cut in APT can be a real burden. In order to deal with this problem, very often, turning-macros have been developed (see also chapter 3.2.).

Most highly automated two-axis NC-languages (such as COMPACT II, MI/GETURN, EXAPT II, etc...) have been provided with system-macros that are able of roughing and finishing shafts from a raw material barstock or forging. This possibility (flexibility), however, is non-existant in APT.

In the author's opinion, the input format should be an easy-to-use tool. This infers that the input should be as

concise as possible. This is especially important when describing a part for production purposes. Arising from this feature, the decision was made to utilize a system of incremental volume-elements to describe rotational parts. This basic approach is consistent with the way in which production engineers visual components as an addition of elements rather than an overall shape which the designer may visualize. The merits of this approach lay in

- the structuring of the system, because macros can be written for each particular volume-element, thus decreasing the computation required.
- the amount of data which must be input.
  Once the type of element (e.g. external shoulder or facing) has been defined, only the amount of data required for that element needs to be entered.

In order to allow the reader to appreciate the meaning of incremental volumes, a typical component, split in volumes is depicted in both Fig. 3.3.23. and Fig. 3.3.24. Having decided to develop a system based on volumes, two sets of turning macros (both represented in the drawings) were created for this thesis. In both sets the toolpaths are generated straight from the definition of the volumes. One set serves for barstock turning and the other for forgings. The structure of these sets of macros is less sophisticated than the milling macros and consequently their implementation less general.

As much as a series of 20 subsequent decreasing diameters can be handled by both macros. This capacity can be extended over 20 by intervening in the "RESERV/" statement of the particular APT macro. Both macros have a four-stroke-loop as basic geometry pattern of the cutterpath. This type of machining is commonly encountered in bar turning work and it consists of the following toolmotions (Fig. 3.3.25.) :

1) a positioning motion where the tool is placed in the correct position to take the cut.
2) the cutting stroke along the main drive surface (in negative X - direction).
3) the cutting motion along the previous checksurface (now drive surface) until the tool has retracted a distance of "DP" in the negative Y direction.
4) return of the tool to a convenient position to start the next positioning motion.

The disadvantage of APT only recognizing full circles and infinite - length straight - lines created the introduction of the auxiliary input variable KC(I). Analogue to its function in milling macros, KC(I) denotes the concavity of a circular checksurface. Without this additional information, a high probability of faulty four - stroke - loops would occur (Fig. 3.3.26.). In case of small convex circle - arcs the cutter first cuts (in negative X - direction) until the center of the circle is reached and subsequently the tool continues to machine until the actual surface is reached. Again, the part programmer needs to be extremely careful when dealing with large convex circle - arcs. Indeed, the ordinary procedure

needs to be pursued in case the center of the circle in question lays to the right of the initial cutter position. However, when employing the macros described in thir work, the user needs not to worry as the above mentioned safety algorithms are built-in.

The first turning macro, dealt with in this thesis, "ROUGH" (Fig. 3.3.23.), can produce any configuration of shafts (excluding undercuts and grooves). The second turning macro, "SUPV1" (Fig. 3.3.24.), treats every volume-element (external shoulder or facing) as if it were situated in the fourth quadrant (of the XY-plane) adjacent to the origin. A "TRACUT" statement transforms all the CL data to their actual location in the XY-plane. The creation of this "forging" macro resulted from the fact that the machining of a stepped shaft out of a forging is a very frequent occuring event in a machine shop.

Several auxiliary macros to "ROUGH" and "SUPV1" have been written, for instance to calculate the number of passes, required to machine each volume-element. The basic algorithm on which the multiloop systems are based are built-in in the flowcharts, given in appendix 1.

In the turning version of COMPACTII [7], there are two avenues of approach open for the user. The fast turn cycle (Figure 3.3.27.B.) is mostly preferable over the normal turn cycle (Fig. 3.3.27.A.) as it has the power of cutting a series of interconnected diameters. APT, too, in connection with the above developed turning systemmacros, provides the part programmer

with both systems. Indeed, due to APT's versatile section III editing facility, complex premachined shafts can be turned (Fig.3.3.28.)

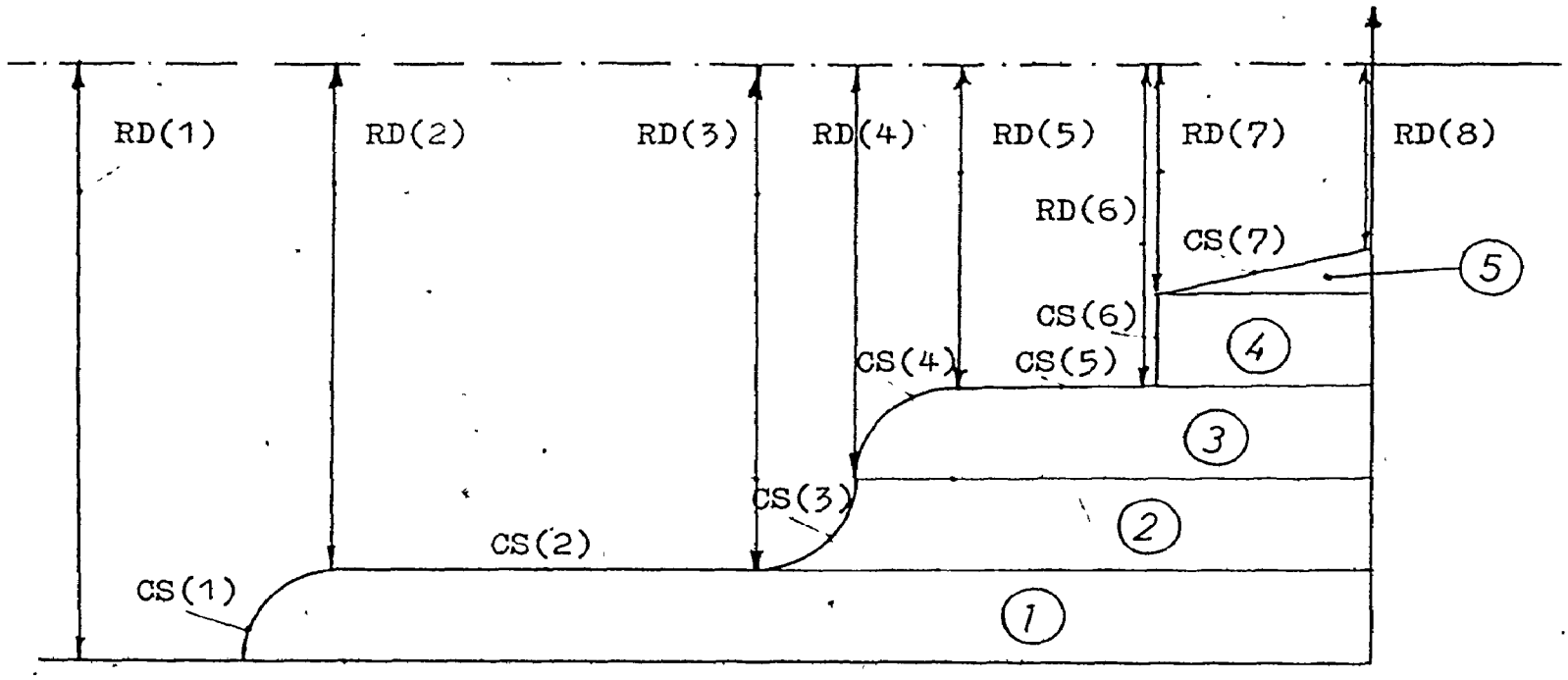


Fig. 3.3.23.
Example of a typical component machined by bar turning (application of "ROUGH" macros)

Fig. 3.3.24.
Example of machining a stepped shaft out of a forging (handled by "SUPV1")

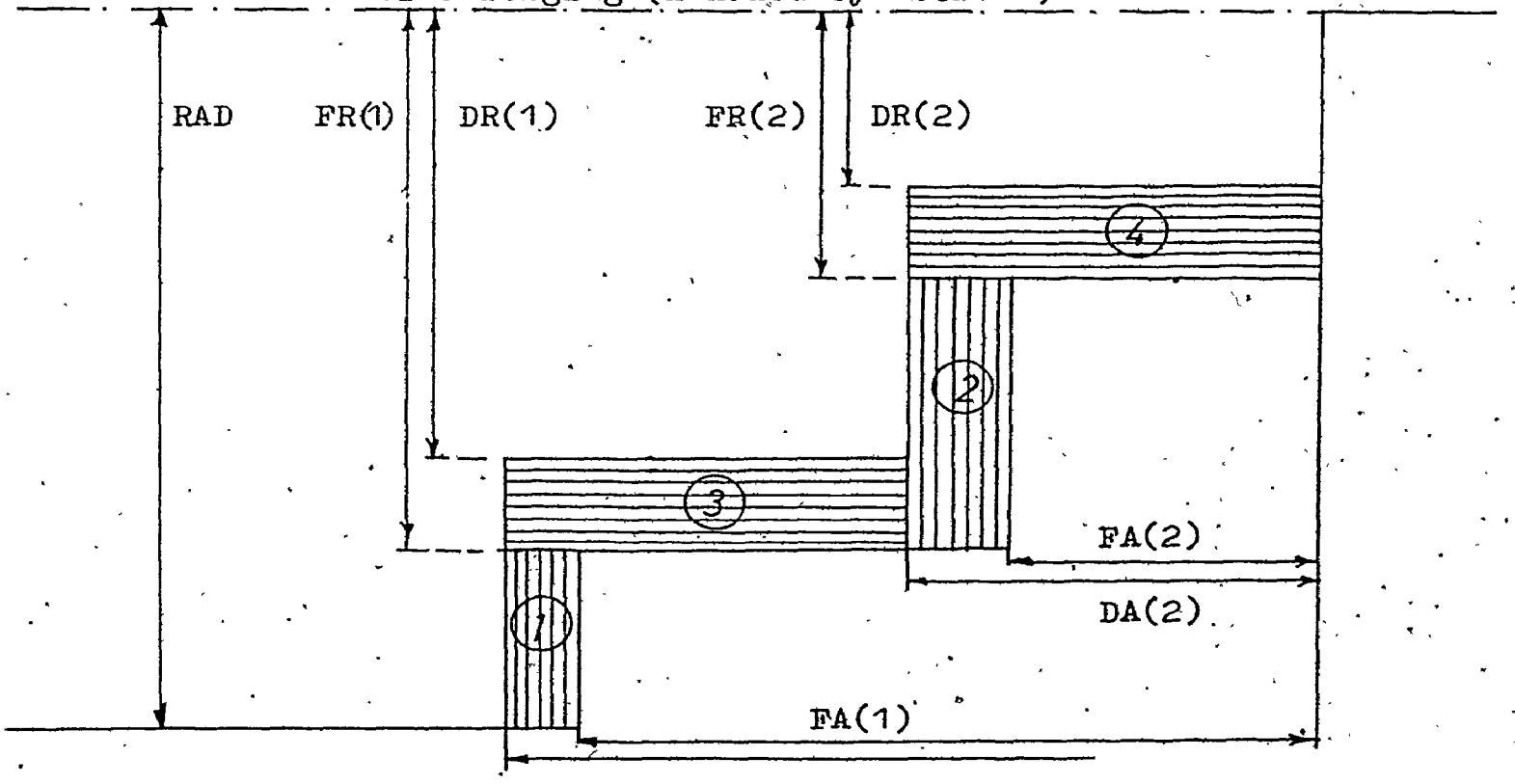


Fig. 3.3.25.

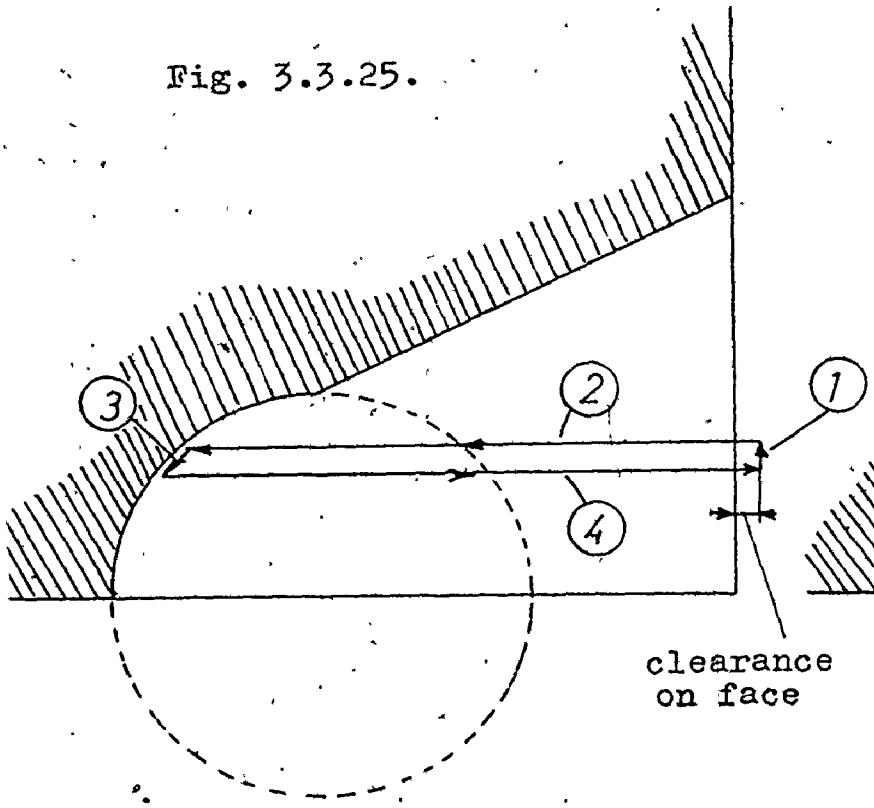
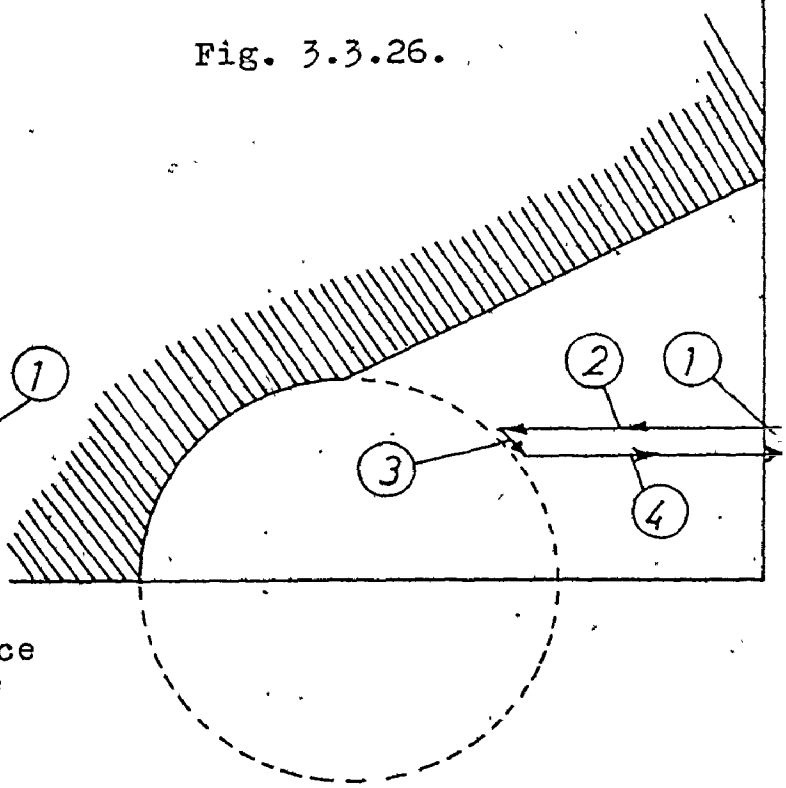


Fig. 3.3.26.



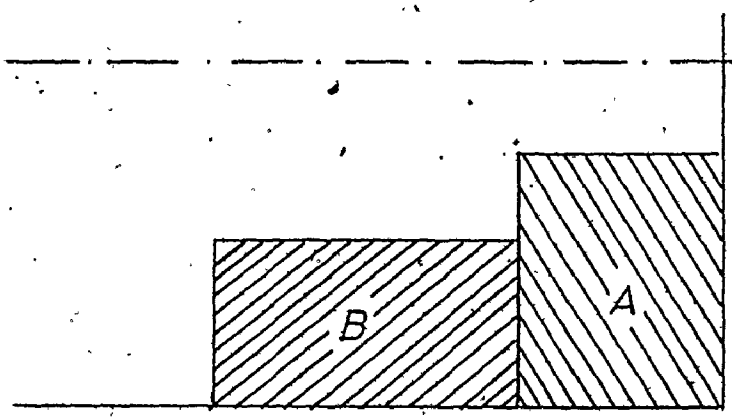


Fig. 3.3.27.A.

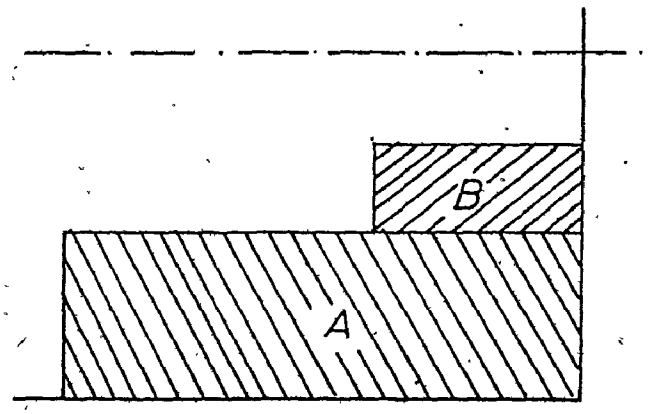


Fig. 3.3.27.B.

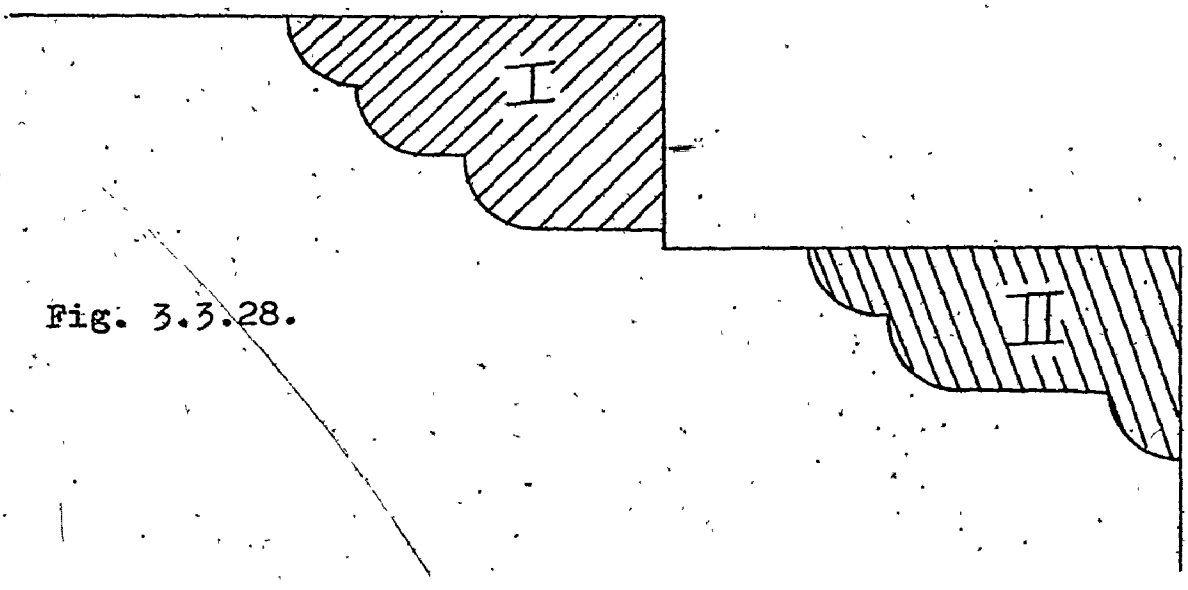


Fig. 3.3.28.

3.4. The advantages of preprocessing.

3.4.1. General. [6.,40]

The statement that any NC machine - tool only operates as faultlessly and efficiently as the pertinent program punched on tape directs it to do, is not a mere repetition of an obvious truism. It is of paramount importance to obtain reliable control tapes. A sequence of trial runs of the tape on the actual machine is only acceptable for cases where NC is used for complicated workpieces in relative large quantities or for the production of uniquely difficult key - parts which, without the application of NC, could not be produced at all. Indeed, machining testpieces, subsequently removing errors, and optimizing the program both kinematically and technologically in order to render the program reliable, can more or less be tolerated in both cases, mentioned above. However, it cannot be justified in an environment where the NC machine is employed as a means of automation in the everyday production of usual workpieces in small lots. In the latter case, the initial tape must be of a consistently high quality, if the company is to capitalize fully from the short set - up times.

The reliability of computer - aided programming of NC machine - tools may be analysed from the following viewpoints : hardware, system - software and part programming. The reliability of hardware is obviously of great importance, however this is outside the scope of this work. The discussion of system - software's reliability certainly must include the productivity of producing reliable software. An interesting Norwegian

paper[40] states that the software costs will soon count for a major part of the total cost of a manufacturing system. Estimates made among others by the Japanese MUM-group indicate a distribution of 70-30 between software and hardware costs in future manufacturing systems. For instance, the US Airforce System Command, certainly an established and experienced institution, incurs yearly an estimated expenditure of nearly $ 1,500,000,000 (1.5 billion dollars) for computer software alone. [6]

Some misconceptions exist about software in general and NC programming system software in particular. Much attention is devoted to theoretical performances of particular systems, often dealing in great detail with optimization of some specific factors. Sometimes this is presented in such a way, as if converting logical flowcharts into instructions for the computer, were merely routine. Unfortunately, it does not quite work that way, and anyone who has not the practical experience of using a computer, will be quite surprised by the amount of time and effort that is still needed at that stage, before an industrially usable and consequently extremely expensive software package becomes available. Unfortunately, only few quantitative data are available in the literature, concerning NC software. The MI/GETURN processor, as an example of a comprehensive, highly automated NC system, is written in Fortran IV and contains 25000 statements. The more versatile and powerful APT system took more than 100 man-years to develop and needs at least a 256 K memory computer.

A software package of any appreciable size with no errors whatsoever does not exist. For example, the most thoroughly tested software ever, was probably that of the Apollo manned spaceflights. Yet, quite a few spectacular software failures, luckily without serious consequences, occured during actual missions.

The software systems, that the NC design of this thesis is based on, are the APT system, the supporting APT system macros and the preprocessor minicomputer software (see later). For nearly two decades the APT system has been tested out and improved. The latter two software packages have also been thoroughly debugged during the course of this work and tried out on a multitude of configurations. Notwithstanding all this, the basic difficulty remains, that any amount of testing can only reveal the presence of errors, but not their absence.

The problem of part programming reliability may be considered as one of the keyfactors in the NC application in general. The number and frequency of part programming errors depend primeraly upon the concentration, motivation and competence of the part programmer. The NC design, developed in this thesis, effectively helps to evade errors, or, if already committed, it detects errors as much as possible. This design is based on the existing APT system, improved by system macros (see chapter 3.2.) and preprocessing support. Only the latter will be analysed further in this chapter.

3.4.2. Implementation of preprocessing in the NC design. [2,24,30]

The computer programs, that constitute the preprocessor, have been found convenient and useful in generating part programming-text, in particular an APT part program. Our objectives are simple : to save money by reducing part programming cost and to upgrade the existing APT system to a highly-automated system containing versatile systemmacros and capable of determining technological data as a part of an integrated information-processing system. The preprocessor fundamentally consists of four distinct functions :

1) Verification of input geometry
2) Calculation of cutting conditions and cutter radius for milling (technology)
3) Calling up of macros
4) Editing of the part program

All of these functions are executed on a minicomputer.

Before analysing these various functions, the reader should realise that the preprocessor greatly simplifies the part programming and consequently substantially reduces the part programming costs. The part programmer, only being human, is bound to make programming errors. Syntax errors, which include violations of the grammar (such as incorrect spelling of major or minor words in APT statements), could practically all be detectable by a diagnostic preprocessing of the part program. Logical errors arise when formally correct APT statements contradict among themselves or exceed limitations imposed by geometry, technology or programming system. A comprehensive preprocessing system should include a proportionate degree of diagnostic analysis to find the detectable errors before the actual proces-

sing begins. How this is realised in practice will become clear in the following description of our "preprocessing tool".

As briefly mentioned in chapter 3.3., a minicomputer [30] was found to be very succesful as preprocessing tool. The PDP 11/34, that the author had access to in the Mac Master University CAD and CAM laboratories, is a stored program general purpose digital minicomputer. In its actual configuration, this stand-alone minicomputer has a core memory of 16 K (16 bit words), a hard copy terminal (type LA36 DECwriter), a dual device (RX11) floppy disk unit, a RK 05 fast disk drive and a GT46 graphics display. Over the last years we have noticed an introduction of a large number of lowpriced physically small computers. These so-called minicomputers have enjoyed a tremendous success in the market for two reasons : low price and flexibility. The wide use of integrated circuits in today's computers has brought the computer price down. The flexibility of a general purpose computer has never been doubted, for it is established as a general purpose scientific tool. So it is not surprising that NC has also turned to the minicomputer. Besides part programming preparation and verification, another example is the trend of the lowcost minicomputer taking over the place of a special hard-wired controller for NC control functions. Indeed, the minicomputer has been found to be an attractive and feasible approach to computerized NC.

The language selected to run on the minicomputer was BASIC, which is rapidly becoming as popular as Fortran. Not only is BASIC easy to learn, but it is also interactive. This means that the computer and the user can carry on a two-way conver-

sation via the typewriter or via the CRT. The user's program can direct the computer to print questions, thereby requesting and labeling new data. Another advantage is that each entry is examined as it is made. If the user should answer a request for a new value by typing a non-digit or by an unreasonable value, the computer can reject that input, print an errormessage, and repeat the question. The author has established a software set that especially exploits this facet of the BASIC language in order to prevent gross errors when running the programs. One must admit that this BASIC feature greatly reduces debugging time when writing software.[24]

It is necessary that such errors are not only detected, but also that their location is properly indicated to the part programmer. This offers the user the oppurtunity to correct them. One should realise that we have come a long way from the diagnostical features in manual programming. Apart from the parity check on the punched tape, there is no inherent diagnostical properties in the coding system for hand-programming. This is why tape verification requires considerable efforts, often occupying the actual machine-tool itself.

In order to clearly show the economical aspects of preprocessing, the following quantitative illustration is presented to the reader. In NC programming systems with a high degree of preprocessing, a great percentage of incorrect programs has been stopped early enough because most diagnostical checks are carried out before any processing of the input data has begun. Needless to say that this tremendously limits the costs of abortive runs. Fig. 3.4.1. shows the influence of preprocessing in

MI/GETURN, an example of a highly automated NC turning language. [2]

In this work the author has attempted to show that, with a reasonable cost expenditure, preprocessing, including automated technology, contributes decisively towards a high degree of reliability of part programming, while upgrading the existing APT system to a highly automated design. However, compared to very sophisticated NC systems, such as EXAPT and MI/GETURN, which needed hundreds of manyears of research and development, it is obvious that this NC design has not yet reached that high level of development.

As a great deal of energy went into establishing the APT system macros (turning and milling), the development of preprocessor minicomputer programs has been limited to software involving the preparation of 2½ D milling part programs. The user has two dialogue - facilities at his disposal : interactive graphics, using the CRT, and interactive "conversation", using the teletype.

It also needs to be pointed out that tape preparation on a minicomputer is not a cure - all. It doesn't replace the need for a skilled part programmer when elaborate workpieces with complicated contours are required. Indeed, eventhough part programming techniques have improved immensely since the early days, the process of writing a sizeable complicated part program and correcting errors remains a most difficult activity.

The skill needed and the effort required to establish a correct part program when employing the NC system, developed

in this work, lies in fact between the two extremes. In contrast with part programming in a sophisticated NC language, such as APT, a company will discover that even part-time users, who are not trained programmers, achieve to establish a correct part program. In the literature the description can be found of a series of "in-house-developed" NC systems. Part programming in these systems comes down to inserting a variety of dimensions straight from the blueprint into the computer. Anyone who can read a blueprint is actually capable of generating a control tape without spending weeks learning how to program. Implementing our system isn't quite that simple. One should bear in mind, however, that our design is based on one of the most difficult but also most powerful and versatile NC systems. Consequently our design contains all the benefits from this (i.e. power + versatility). To illustrate the simplified input of this NC design, a number of examples will be presented to the reader in chapter 4. It'll be clear to the reader that, instead of using straight APT, where the main effort is put into the syntax, the user may concentrate more on the semantics. The author would like to remark that this design relieves the expert part programmer, as all part programming will not funnel through him anymore, but skills will still be needed for complicated elaborate workpieces.

An economic evaluation of the system can be performed by comparing this NC design with the industrially applied microprocessor approach to NC. The existing microprocessors (e.g. Numeridex 7800, MDSI, BAYERNC9, etc...) including a high speed

reader aswell as a hard copy terminal, can now be purchased for a total of $ 10,000. An industrial application of this new NC design including a 16 K minicomputer, a hard copy terminal, dual drive floppy disks ($ 12,000), a high speed punch, a high speed reader ($ 4,000) and an 2780 emulator ($ 5,000) would add up to $ 21,000.

As far as APT processing on a time-sharing basis via a microprocessor is concerned, the computing costs are twofold. A realistic assessment would reveal that 50% of the computing costs are due to on-line editing and storing aswell as the regular terminal charges. The second source of costs (the other 50%) are imputed to computer runs. The benefits to be gained from the approach in this work are owing to the off-line capabilities of our intelligent terminal. Indeed, a realistic estimate shows that the first sort of computing cost (mentioned above) can be reduced by 75% while the cost involving computer runs probably will be halved. This would approximately amount to a 60% reduction in computer costs per annum. In a typical case of a system supporting two part programmers, an avarage of $ 10,000 could be saved per year. Thus, eventhough one would have a supplementary initial cost of $ 11,000 and also a higher service for a minicomputer compared to a microprocessor, it can easily be deducted that, assuming a five year economic life of of the project, the rate of return of investment will be quite high ($> 50\%$).

The equipment that the author had access to has been depicted in fig. 1.4. It needs to be stated however that in the industrial application of this design the CRT display can be

replaced by a less expensive HP flatbed plotter. Indeed lacking the CRT display in this design would not mean an impossibility of interactive geometry verification. The existing software can economically, via the FINMAC APT macro, generate a contour of the part.

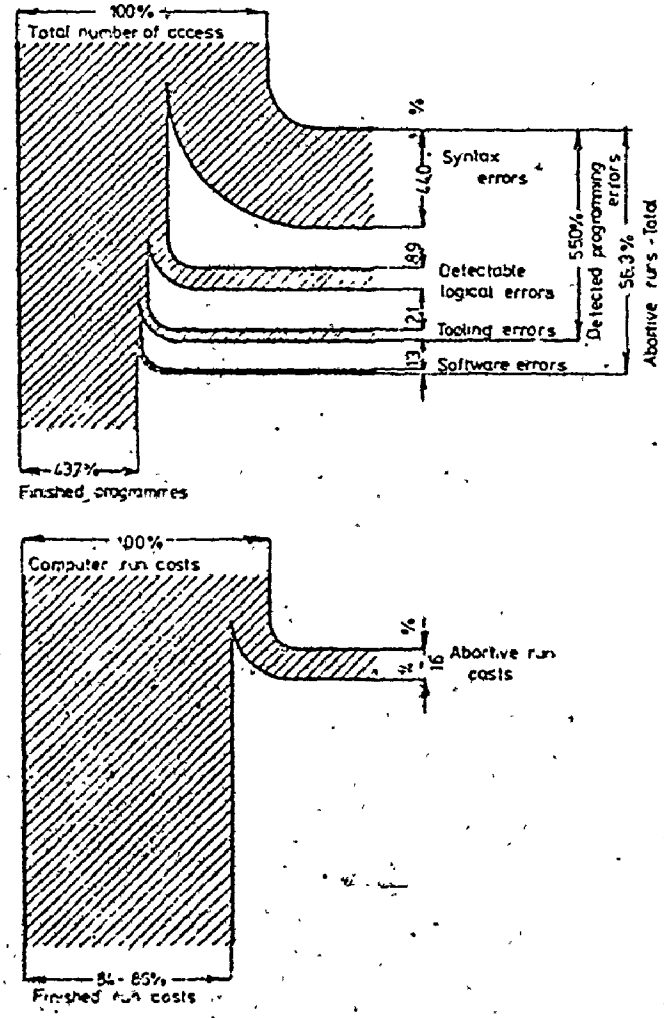


Fig. 3.4.1.

The importance of preprocessing on the part program reliability and the computer run costs when using the MI/GETURN NC system.

2)

3.4.2.1. Verification of geometrical statements. [2,14,15,31,32,33,34]

An effective means to find programming errors that are not detected by syntax and logical checks and that remain even after additional checks, available in some NC systems [2] (e.g. the built-in redundancy check* in MI/GETURN) is plotting or visual display. The aim of the geometrical verification is to indicate major mistakes that are more likely to happen than errors in accuracy which cannot be seen on a plot or on a display.

Recently a large number of researchers have published their achievements which put into practice the trend of integrating the CAD and CAM systems [31]. One author [32] states that from the advent of the first CRT display capable of displaying vectors, the computer industry has been besieged with ideas and invisioned advances concerning the possible uses of a graphic console in a production environment. A multitude of attempts has been made to achieve this goal. [33].

The great importance of using graphics, plots or CRT displays to develop and debug NC programs is now beyond any doubt. The following designs represent a few of the many approaches to interactive part programming which characterizes the NC-scene today. In the "Ecole Nationale Supérieure de Techniques Avancées de Paris" (France) [34], the display of a part on a CRT and interactive text-editing of the MINI-IFAPT source (NC) program on a minicomputer have been previously develo-

* Redundancy check: the reason for this diagnostical feature is that the probability of making two errors instead of one, in addition making these errors so that they compensate each

is

ped with success but unfortunately this system lacks a few features that we did implement in our design. Not only is the system limited to drilling operations, but the permanent programs, comprising the graphical operating system of the computer, occupy almost double the core-size this design uses and, last but not least, there can be no interaction with the part programmer from the graphics display since the system does not offer re-entry facilities. Eventhough the MINI-IFAPT language is confined to only 2½ D machining, the design has the advantage of being able to execute preprocessing, processing and postprocessing on that very same minicomputer.

In contrast with processing in a batch environment, the American APT/IGS* system provides the user with the capability of generating the console tape of the desired part from the graphic console[32]. In an interactive graphic environment, the on-line computer graphics are used to facilitate the part programmers job and meanwhile to reduce the number of trial runs before producing a good part. Indeed, compared to processing in a batch environment where it might take the user up to ten submittals before one error-free run through the APT processor is made, the number of iterations of the batch mode is replaced by a more direct interaction with the graphical terminal. One should bear in mind that the use of interactive programming systems based on a timesharing basis might not be economical since it involves maintaining a terminal to the timesharing system.

* APT/IGS : Automated Programmed Tool/Interactive Graphic Systems

One of the latest achievements which attempts to integrate CAD and CAM, is the Japanese TIPS1 system. The TIPS1 [14] program apparently first optimizes the design (as far as area, volume, weight, center of gravity, moment of inertia, stress, strain, etc... are concerned) and subsequently draws the part and automatically generates the control tape in order to manufacture the desired part. Not only would the analysis of all integrated CAM - CAD systems (e.g. volume - brick technique, [15] etc...), recently published in researchpapers, be exceedingly extensive in volume, but it also needs to be stated that quite a number of these developments, concerning this subject, deal with specific applications. For well defined examples (such as shafts, dies, bearings, gears , ...) or other items that represent a very rigid family of parts, the development of CAD - CAM integrated systems is beyond the scope of academic research.

Part - geometry input verification, via an interactive graphics terminal, can shorten and refine the procedure used to produce a part. Indeed, for highly automated programming systems, such as this design, which process the input semi - automatically (via interaction with the part programmer), both geometrically and technologically, it seems to be more efficient to plot or display the input rather than the output. It appears to be preferable, as is the case with other diagnostical checks, that errors are detected before the actual processing begins. This implies that such highly automated systems produce a correct output if the input was found correct.

This NC design provides a visual indication of the manually-entered data and a means for the part programmer to visually confirm the input or to modify in case of discrepencies and errors. The CRT display which has become an interactive communication and display device, must be capable to permit refreshing because the part programmer must, via the interactive graphics tool, be able to create and modify pictorial information on the spot. Besides a detailed analysis in appendix 1 (flowcharts), the practical realisation of the integration of part-geometry verification in the preprocessor will be described in chapter 3.4.2.3.

3.4.2.2. Inserting of tooling and cutting conditions.

The automation of the technological aspect of CAM* is, beyond any doubt, a development of great economical, technological and human importance. Due to the evolution in data-processing and computertechnology, the stipulation of technological parameters with aid of computer is facilitated greatly. However, it needs to be pointed out that any intention to build up a fully automated technological system should be abandoned. The problem is not only enormous in volume but the final goal is undesirable from human, technical and economical point of view. We have learnt that we have to strive for combinations of man, computer and production equipment in cooperating systems. The use of dialogue systems such as interactive graphics and alphanumerical terminals, where man can use his normal communicationaids, is very important in determining technological data.

In this work the author attempts to show that, as far as "technology" is concerned, it is possible to formulate approximate methods of selecting economic machining conditions for a certain process, provided the knowledge of the geometrical configuration and the knowledge of the pertaining process (e.g. turning, milling, etc..,) are available.

Of this system, still in the development stage, only the technology concerning milling operations has been applied. Under the term "technology" cutting conditions and tool selection are understood. Technology relates to features such as prefered (optimized) machining sequences, toolradius, feed, rota-

tional spindle speed, etc... The milling optimization modules, based upon the theoretical analysis of chapter 2, were written in the BASIC language (minicomputer software) and integrated in the preprocessor software package.

The practical realisation at the minicomputer console is performed as follows. The preprocessor control program calls a technological program into memory (the milling optimization program OPMIL3). Various geometrical and technological data files (stored on floppy disk, disk drive or even casette) first will be searched for on the appropriate peripheral device and subsequently be opened (i.e. attached on-line). The description of the geometry, input during a previous preprocessing sequence, rearranged and stored on the geometrical data files, will now be manipulated by the optimization program (e.g. OPMIL3). During the execution of the latter program, the cutting conditions are computed via interaction with the machinability data, operation requirements, tooling available, etc..., stored on the technological data files.

The optimized cutting parameters will not be imposed upon the user but will appear in front of him on the console in tabular format. The part programmer, after letting his workshopexperience and human common-sense digest the computer output, selects himself the cutting parameters and toolspecification from the tabels on the teletype listing or from the tables depicted on the CRT display. A clarifying illustration of how technology is inserted during the part program preparation will be presented in an example in chapter 4.

As mentioned earlier, only the milling optimization routines (see chapter 2) have been implemented in this work and integrated in the preprocessor software. The extension of this approach to selecting economic machining conditions in case of a turning process has not yet been realised. It should, however, be pointed out that research concerning multiple turning operation is being pursued at the university. Generating BASIC software to integrate the latter in the now existing preprocessor package could later easily be accomplished by a trained programmer.

The actual experience accumulated over the years is of paramount importance for practical aspects of NC machine-tool exploitation in a company. Due to the modularity of the preprocessor, the user will be able to enter his own "know-how" into the technological files. Through actualization (updating) of the technological data files the quickest possible adaptation to actual manufacturing conditions is achieved. One can conclude that many benefits are to be gained from the modular built-up of the preprocessor as this low-cost system can be tailored to suit specific companies.

3.4.2.3. Calling up systemmacros and editing of the part program.

Because of the continuously decreasing price of minicomputers, the use of interactive programming systems, based on minicomputers, becomes more and more within the means of medium size and small companies.

For this thesis, minicomputersoftware has been established for part program preparation. It consists of the following sets of programs. A first set asks appropriate questions in sequences, which guide the user while he enters his own parameters into the computer. A second set of programs digests the information, obtained by the first set, and, via a number of datafiles, edits the actual part program. This special editing system provides an extremely rapid method of debugging the input. The collection of all these programs constitutes the preprocessor. The preprocessor contains special modules that analyse the information entered by the part programmer. For instance, some modules will recognize input geometry in the APT language, manipulate the different parts of the strings (convert it into canonical form, if not already done by the part programmer explicitely), transform these results immediately into pictorial information, display it on the screen and edit the geometry data files until the part programmer is completely satisfied with the graphical display.

The interaction between the user and the computer may occur in a conversational mode using the keyboard or using the

lightpen sensitive subpictures in a so-called menutechnique. This implies that besides the minicomputer (16 K memory (16 bit words)) the following two components are necessary as interactive input/output devices : a teletype terminal and graphic terminal (the latter is in fact only strictly necessary for the interactive part geometry verification).

Most of the teletypes available today feature an alphanumeric keyboard aswell as function keys which can be used for interactive conversational type of programming. The CRT displays must be of the interactive graphic type (not a passive CRT display) because only the former type enables the user to input data using cursors, lightpens and graphic tablets.

The following will illustrate how various information or logical decisions can be obtained from conversation between the part programmer and an interactive graphics peripheral device. The CRT display (screen) can be subdivided into as many lightpen sensitive subpictures as needed. Any lightpen hit within a subpicture will be recorded and a certain course of action will be taken accordingly (e.g. coordinates of a point where the lightpen hits may be recorded, etc...).

Some of the programs of the preprocessor software package take care of calling up the desired APT macros (and the matching "RESERV/" statement), inserting them in an appropriate spot of the part program. Subsequently, control is transferred to other programs responsible for the final editing of the part program. The actual editing-phase on the minicomputer consists of opening certain files (e.g. geometrical data

files, cutting condition data files, etc...), extracting from those files the necessary information for the part program and last but not least, arranging the various accessories, in order to obtain the final lay-out of the part program. The programmer, subsequently, acquires the APT part program listing on the teletype which automatically indicates the termination of the preprocessing.

As far as reliability of the output part program is concerned, we may say that the edited part program not only contains, almost certain, correct geometrical statements but we can also, with a high probability, proclaim that the toolmotionstatements are right because of the implementation of the systemmacro-concept.

The editing phase of the preprocessor, in which the actual text is generated, is well worth its inclusion in the preprocessor as it avoids the error-prone editing of tapes and carddecks.

CHAPTER 4

Examples of the usage of the NC system. [4,5,8,26,35]

The purpose of this chapter is the simple demonstration of the capabilities of the system. Three examples will be presented. The first example illustrates the part program generation of a composite part on the PDP 11/34 minicomputer via the preprocessor software package. The second and third example deal with control tape generation via the APT processor and respectively a NC lathe and a NC milling machine postprocessor.

Example 1.

The part program of a composite part (Fig. 4.1.), consisting of two pockets of different depths of cut, will be established by the preprocessor software package via a part programmer - teletype - CRT display three - way conversation. Since the macro is stored in the APT system library (when supported by APT IV), the amount of programming that the part programmer must carry out, is reduced to a minimum. Besides inputting the geometrical statements, the only task that is left him to perform is inserting a number of values.

In this example the user typed in a "N" (i.e. No) in reply to the question "Are you satisfied with the (displayed) geometry ?". He subsequently reenters the geometry he wants to modify. Automatically, due to the refreshed CRT display, the new geometry is being displayed (Fig. 4.2. and Fig. 4.3.) and the above mentioned question will keep on appearing on the teletype until the user is completely satisfied and wants to continue the "conversation".

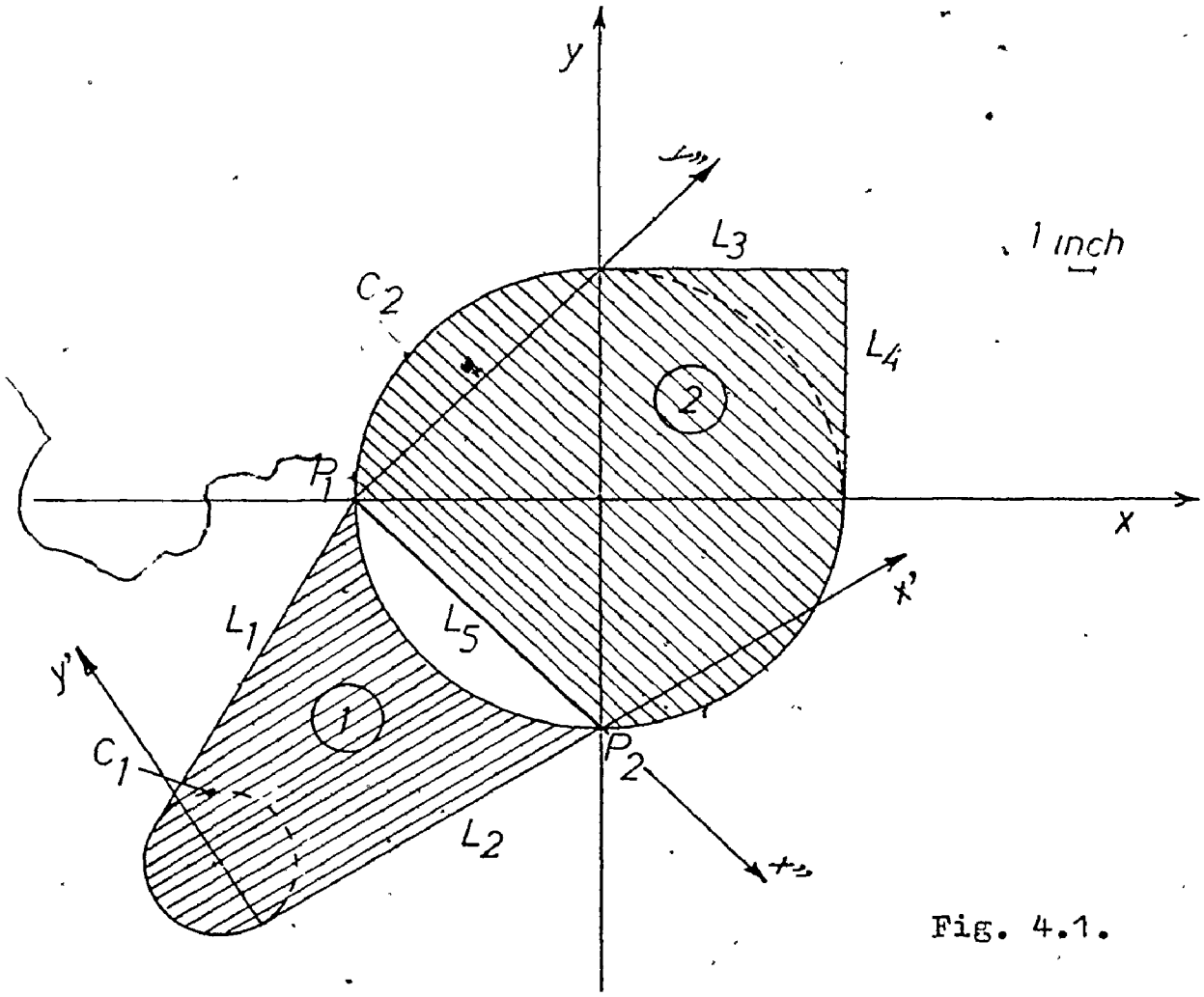


Fig. 4.1.

As shown in the flowcharts, only two point-, three line- and two circle- APT - definitions can be employed to input geometrical statements. Eventhough the names of the boundary surfaces (also called checksurfaces) of the pocket(s) ought to be CS(I), the user can enter any name he wishes to.

When generating the part program for a turned part, the user should bear in mind that the simplicity of the stepped shaft approach allows him to shortcut the normal input procedure and enter the preprocessor not in APTØ, as is generally done, but in program APTURN or program BEN1. In the latter case, the part program will be created interactively with the CRT display.

As far as the generation of milling part programs in a "conversational" mode with the CRT display is concerned, the con-

versation between the user and the milling optimization program has not yet been established on the screen. The interaction completely occurs on the teletype. The optimization routine, so far, only works with one set of parameters, such as a certain maximum power (e.g. 10 hp), stressconstraint (e.g. 180000 psi), etc... However, anticipating future extensions, files have been created that contain the necessary input parameters for the optimization routine. Software has also been written in order to modify or update the existing files in an interactive way via a lightpensensitive CRT display.

During the preprocessing the user might come across the following concepts. The "Relieve height" is the height (in inches) above the Z = 0 plane at which the cutter can travel in rapid traverse mode without any danger for collision. The "Angle of rotation" represents the angle at which the coordinate system needs to be rotated in order to exercise the first cut in the positive X - direction. The angle increases positively counterclockwise from the positive X - axis.

The "geometric language" created for describing the pocket might at first seem awkward to the reader but the philosophy behind it has proven to be successful in various highly automated two axes languages. In the EXAPT II language, the part programmer contours the blank and the finished part in order to determine the material that ought to be removed ("CONTUR" statement) [36]. When defining the pocket, one should bear in mind to number the boundary surfaces clockwise starting from the lowest intersection point. The contouring statements are in a straight-forward APT-like language. The following pages illustrate how the part programmer would generate the part program of a typical composite part via the preprocessor package.

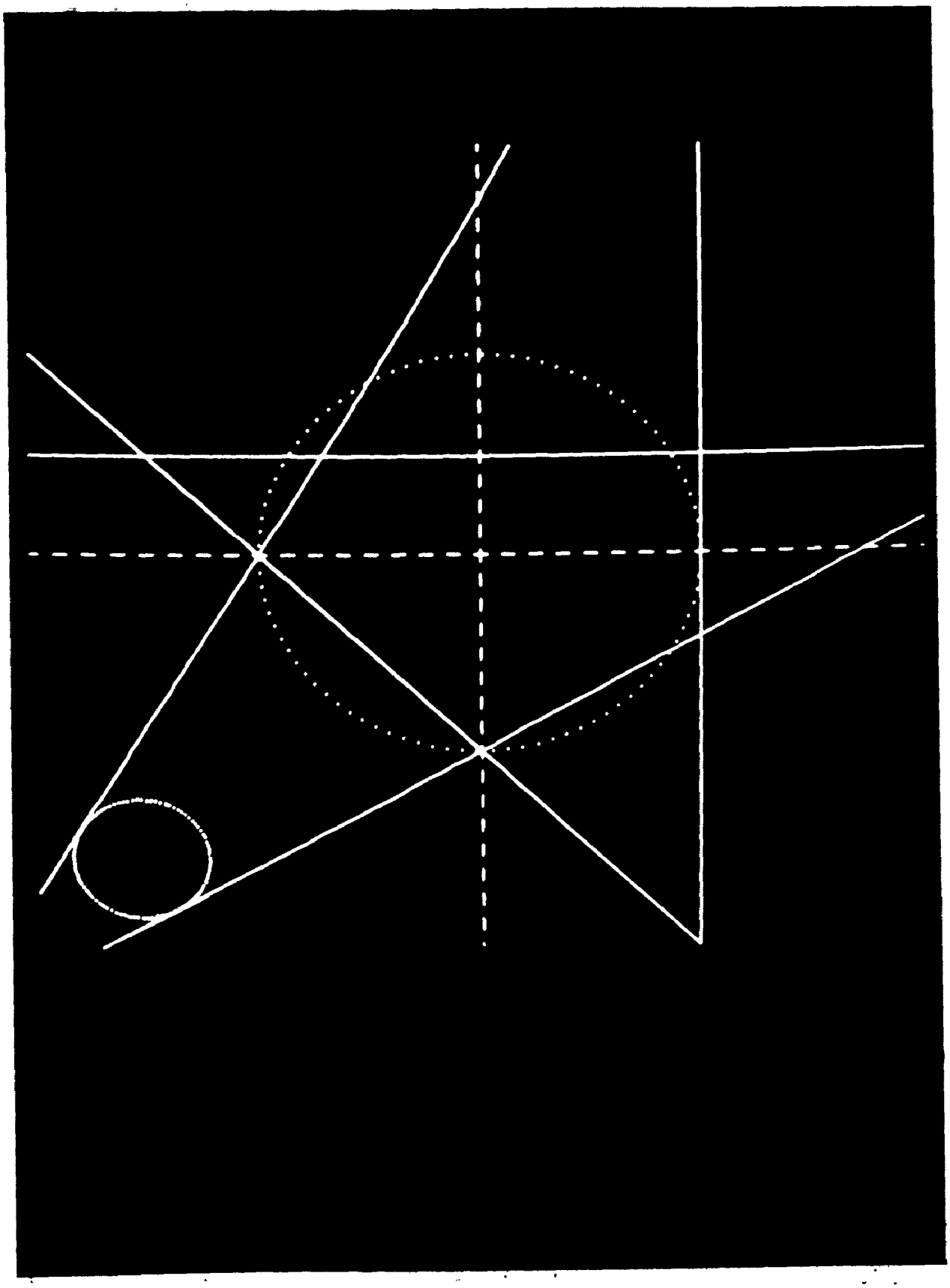


FIGURE 4.2 INCORRECT GEOMETRICAL INPUT

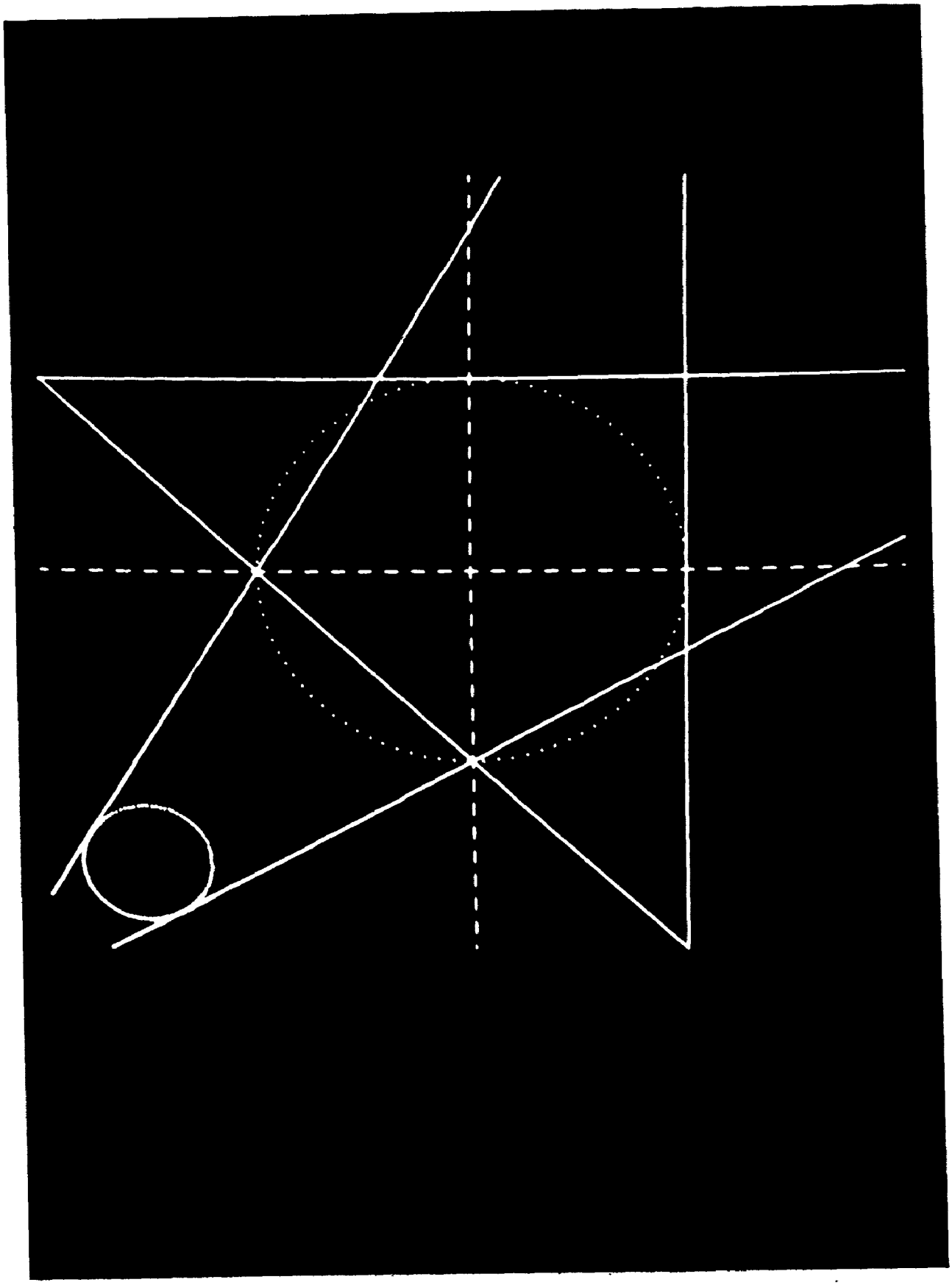


FIGURE 4.3 CORRECT GEOMETRICAL INPUT

```
APTO        BASIC V4.1B-02

PREPROCESSOR SOFTWARE BAC 40-F ( PART PROGRAM PREPARATION )

EXTREME VALUES OF THE SCREEN ARE XO,YO,X1,Y1
?-20,-20,20,20

INPUT ALL THE GEOMETRICAL STATEMENTS
          LAST ONE IS : END
?P1=POINT/-10,0,0
?P2=POINT/0,-10,0
?L1=LINE/P1,ATANGL,60
?L2=LINE/P2,ATANGL,30
?C1=CIRCLE/XLARGE,L1,YLARGE,L2,RADIUS,3
?L3=LINE/CANON,0,1,0,5
?L4=LINE/CANON,1,0,0,10
?C2=CIRCLE/CANON,0,0,0,0,0,1,10
?L5=LINE/P1,P2
?END


**********************************
GEOMETR. DEF. OF PART PROGR. (VERSION 1)
**********************************


P1=POINT/-10,0,0
P2=POINT/0,-10,0
L1=LINE/CANON,-.866025,.500001,0,8.66025
L2=LINE/CANON,-.5,.866026,0,-8.66026
L3=LINE/CANON,0,1,0,5
L4=LINE/CANON,1,0,0,10
L5=LINE/CANON,.707107,.707107,0,-7.07107
C1=CIRCLE/CANON,-15.4641,-15.4641,0.0,0,1,3
C2=CIRCLE/CANON,0,0,0,0,0,1,10
FINI

**********************************
GEOMETR. DEF. OF PART PROGR. (VERSION 2)
**********************************


P1=POINT/-10,0,0
P2=POINT/0,-10,0
L1=LINE/P1,ATANGL,60
L2=LINE/P2,ATANGL,30
C1=CIRCLE/XLARGE,L1,YLARGE,L2,RADIUS,3
L3=LINE/CANON,0,1,0,5
L4=LINE/CANON,1,0,0,10
C2=CIRCLE/CANON,0,0,0,0.0,1,10
L5=LINE/P1,P2
FINI


ARE YOU SATISFIED WITH THE GEOMETRY (Y OR N) ?N
HOW MANY GEOMETRICAL STATEMENTS DO YOU WANT TO MODIFY ?
?1
INPUT THE NEW GEOMETRICAL STATEMENTS
?L3=LINE/CANON,0,1,0,10
```

```
**********************************************
GEOMETR. DEF. OF PART PROGR. (VERSION 1)
**********************************************

P1=POINT/-10,0,0
P2=POINT/0,-10,0
L1=LINE/CANON,-.866025,.500001,0,8.66025
L2=LINE/CANON,-.5,.866026,0,-8.66026
L3=LINE/CANON,0,1,0,10
L4=LINE/CANON,1,0,0,10
L5=LINE/CANON,.707107,.707107,0,-7.07107
C1=CIRCLE/CANON,-15.4641,-15.4641,0,0,0,1,3
C2=CIRCLE/CANON,0,0,0,0,0,1,10
FINI

**********************************************
GEOMETR. DEF. OF PART PROGR. (VERSION 2)
**********************************************

P1=POINT/-10,0,0
P2=POINT/0,-10,0
L1=LINE/P1,ATANGL,60
L2=LINE/P2,ATANGL,30
C1=CIRCLE/XLARGE,L1,YLARGE,L2,RADIUS,3
L3=LINE/CANON,0,1,0,10
L4=LINE/CANON,1,0,0,10
C2=CIRCLE/CANON,0,0,0,0,0,1,10
L5=LINE/P1,P2
FINI

ARE YOU SATISFIED WITH THE GEOMETRY (Y OR N) ?Y
DO YOU WANT TO INPUT THE VARIOUS INFORMATION WITH
          INTERACTIVE GRAPHICS (Y OR N)  ?N

WHAT KIND OF OPERATION (MILLING(M) OR TURNING(T) ) ?M



NUMBER OF POCKETS ?2
DEPTH OF CUT IN POCKET  1 ?.2
NUMBER OF CHECKSURFACES IN POCKET  1 ?4
ROTATION ANGLE OF POCKET  1 ?30
```

RADIUS (IN)	FEED (IN/MIN)	VEL. (FPM)	COST/IN^2 ($/IN^2)	TORQUE (P-I)	POWER (H.P.)	STRESS (PSI)	
.1	30.7551	180	.0773508	33	1.85	179999	$
.124	32.8954	161	.0580698	62	2.45	180000	$
.125	36.3918	177	.0520993	63	2.74	179999	$
.126	36.4335	176	.0515598	65	2.76	180000	$
.2	34.8359	151	.0340525	183	4.19	133199	
.24	29.4119	153	.0335123	220	4.25	93867	
.25	35.447	144	.026639	305	5.33	115681	
.26	34.0837	144	.026586	317	5.33	107241	
.3	29.7429	145	.0264185	366	5.37	81268	
.4	22.46	146	.0261988	488	5.4	46371	
.49	18.4594	147	.0261281	598	5.44	31142	
.5	24.0184	130	.0195725	916	7.22	44894	
.51	23.5475	130	.0195787	935	7.22	43180	
.6	20.0153	130	.0196417	1100	7.22	31352	
.7	17.025	129	.0197188	1283	7.16	23126	
.8	14.8969	129	.0197981	1466	7.16	17759	
.874	13.5307	128	.019856	1602	7.11	14905	
.875	16.4837	117	.0163505	2139	8.66	19828	
.876	16.4649	117	.0163512	2141	8.66	19783	
.9	16.0259	117	.0163701	2200	8.66	18752	
1	14.301	116	.0164449	2444	8.59	15217	

$: FEED LIMITED BY MAX. STRESS.

DEPTH OF CUT IN POCKET 2 ?.7
NUMBER OF CHECKSURFACES IN POCKET 2 ?5
ROTATION ANGLE OF POCKET 2 ?-45

RADIUS (IN)	FEED (IN/MIN)	VEL. (FPM)	COST/IN^2 ($/IN^2)	TORQUE (P-I)	POWER (H.P.)	STRESS (PSI)	
.1	16.6947	178	.142295	65	3.52	179999	$
.124	15.9647	171	.120001	99	4.17	179999	$
.125	17.5675	188	.108028	101	4.63	179999	$
.126	17.5938	188	.107269	103	4.67	179999	$
.2	17.7235	155	.0668435	318	7.47	179999	$
.24	18.1602	141	.0542904	515	9.18	179999	$
.25	18.8571	143	.0497775	575	10	179999	$
.26	18.1319	133	.0482508	640	10	179999	$
.3	15.7143	104	.043565	948	10	179999	$
.4	11.7857	77	.0396708	1711	10	145063	$
.49	9.62099	77	.039604	2096	10	99619	$
.5	9.42857	51	.0361839	3208	10	143889	$
.51	9.2437	51	.0361874	3272	10	138650	$
.6	7.85714	51	.0362234	3850	10	102067	$
.7	6.73469	51	.0362351	4491	10	76113	$
.8	5.89286	51	.0362823	5133	10	58921	$
.874	5.39392	51	.0362825	5608	10	49687	$
.875	5.38775	45	.0356027	6302	10	55648	$*
.876	5.38161	45	.0356075	6302	10	55462	$*
.9	5.23809	47	.035755	6302	10	51237	$*
1	4.71429	52	.0363537	6302	10	57605	$*

$: FEED LIMITED BY MAX. STRESS.
@:VELOCITY LIMITED BY MAX. POWER.
*: FEED LIMITED BY MAX. TORQUE.

```
WHICH CUTTERRADIUS (IN) DO YOU CHOOSE FOR POCKET  1 ?.875
WHICH FEED (IN/MIN) ?16
WHICH VELOCITY (FT/MIN) ?117
WHICH CUTTERRADIUS (IN) DO YOU CHOOSE FOR POCKET  2 ?.875
WHICH FEED (IN/MIN) ?5
WHICH VELOCITY (FT/MIN) ?45



GENERAL INFORMATION

INNERTOLERANCE (IN) ?.0005
OUTERTOLERANCE (IN) ?.0005
FEED FOR PLUNGING (IN/MIN) ?10
FEED FOR RETRACTING (IN/MIN) ?20
RELIEVE HEIGHT (IN) ?1
STARTING X COORDINATE (IN) ?0
STARTING Y COORDINATE (IN) ?0
STARTING Z COORDINATE (IN) ?5
WHICH MILLING METHOD (RGH1,RGH2 OR SPIRAL)  ?RGH2

EXAMPLE OF CONTOURING : GOFWD/L1,INTOF,YLARGE

CONTOUR POCKET  1
?GOFWD/C1,INTOF,XSMALL
IS THIS SURFACE CONCAVE (-1) OR CONVEX (+1) ?1
?GOFWD/L1,INTOF,XSMALL
?GORGT/C2,INTOF,XSMALL
IS THIS SURFACE CONCAVE (-1) OR CONVEX (+1) ?-1
?GORGT/L2,INTOF,XLARGE

CONTOUR POCKET  2
?GORGT/C2,INTOF,XSMALL
IS THIS SURFACE CONCAVE (-1) OR CONVEX (+1) ?1
?GOFWD/L3
?GORGT/L4,INTOF,XLARGE
?GOFWD/C2,INTOF,XLARGE
IS THIS SURFACE CONCAVE (-1) OR CONVEX (+1) ?1
?GORGT/L5,INTOF,XSMALL
INPUT YOUR IDENTIFICATION OR PARTNO STATEMENT ?EXAMPLE PART
```

```
8888888888888888888888888881888
     PART PROGRAM (POCKET)
8888888888888888888888888888888

PARTNO EXAMPLE PART
MACHIN/CINAC1-300,CIRCUL,1,0
RESERV/CS,9,KS,9,N,9,G,9,F,9,DUM1,9,DUM2,9,DUM3,9,DUM4,9,$
X,9,Y,9,T1,9,T2,9,A1,9,A2,9,W,9,M,9,CA,9,D,9,$
Q,9,X1,9,Y1,9,X2,9,Y2,9,$
U,50,V,50
IT= 5.00000E-04
OT= 5.00000E-04
FDW= 10
FUP= 20
A= 1

CALL/INIT0,A=IT,B=OT

B= .2
K1= 4
AG= 30
R= .875
FD= 16
RP= 255

KS( 1 )= 1
KS( 2 )= 0
KS( 3 )=-1
KS( 4 )= 0

M( 1 )=-1
M( 2 )=-1
M( 3 )= 2
M( 4 )= 2

N( 1 )=-1
N( 2 )=-2
N( 3 )=-2
N( 4 )=-2

CALL/INIT1,A=AG

CS( 1 )=CIRCLE/XLARGE,L1,YLARGE,L2,RADIUS,3        $$C1
CS( 2 )=LINE/P1,ATANGL,60        $$L1
CS( 3 )=CIRCLE/CANON,0,0,0,0,0,1,10        $$C2
CS( 4 )=LINE/P2,ATANGL,30        $$L2
CALL/INIT2,KB=K1

FROM/(POINT/ 0 , 0 , 5 )

TRACUT/NO

CALL/FINMAC
CALL/ROH2
PRINT/3,ALL

TRACUT/NOMORE
```

```
B  .2
N1  5
AG -45
R- .875
FD- 5
RP= .3

KS( 1 )- 1
KS( 2 )= 0
KS( 3 )= 0
KS( 4 )= 1
KS( 5 )= 0

M( 1 )= 2
M( 2 )=-1
M( 3 )= 2
M( 4 )=-1
M( 5 )= 2

N( 1 )=-2
N( 2 )=-2
N( 3 )= 0
N( 4 )=-1
N( 5 )=-1

CALL/INIT1,A--AG

CS( 1 )=CIRCLE/CANON,0,0,0,0,0,1,10      $$C2
CS( 2 )=LINE/CANON,0,1,0,10      $$L3
CS( 3 )=LINE/CANON,1,0,0,10      $$L4
CS( 4 )=CIRCLE/CANON,0,0,0,0-0,1,10      $$C2
CS( 5 )=LINE/P1,P2      $$L5
CALL/INIT2,BB=N1

TRACUT/M0

CALL/FINMAC
CALL/RGH2
PRINT/3,ALL

TRACUT/NONONE


CALL/FINAL

END
FINI
```

Example 2.

The component to be machined is depicted in Fig. 4.4. The input to and the output from the APT system are given on the pages which follow. The nature of the output gives a fairly good impression of the amount of work that would have been involved in the detailed calculations of the motionstatements. Fortunately, this is all taken care of in the APT systemmacros, developed in this work.

This example deals with the machining (roughing) of a turned part from a bar stock shaft (blank). While finishing the part, the user should bear in mind that there is a small excess of material at the intersection of certain boundary surfaces. Indeed, the toolmotion statement ((1) in Fig. 4.5.)

```
TLLFT,GOLFT/CS(I),PAST,(LINE/PARLEL,LX,YSMALL,B)
```

which is a part of the four-stroke-loop, cannot be implemented in the first machining cycle of a section because the possibility exists that the previous boundary surface CS(I-1) would be undercut. The following toolmotion statements have been inserted instead ((2) in Fig. 4.5.) :

```
TLLFT,GOLFT/CS(I),TO,(LINE/PARLEL,LY,XSMALL,X)
TLLFT,GOLFT/(LINE/PARLEL,LY,XSMALL,X),PAST,
                    (LINE/PARLEL,LX,YSMALL,B)
```

Eventhough a small amount of material is left uncut, the size of the cusp does not require an additional semifinishing pass.

Figure 4.6., figure 4.7. and figure 4.8. respectively represent the part program itself, a section IV hard copy output and a section IV plot.

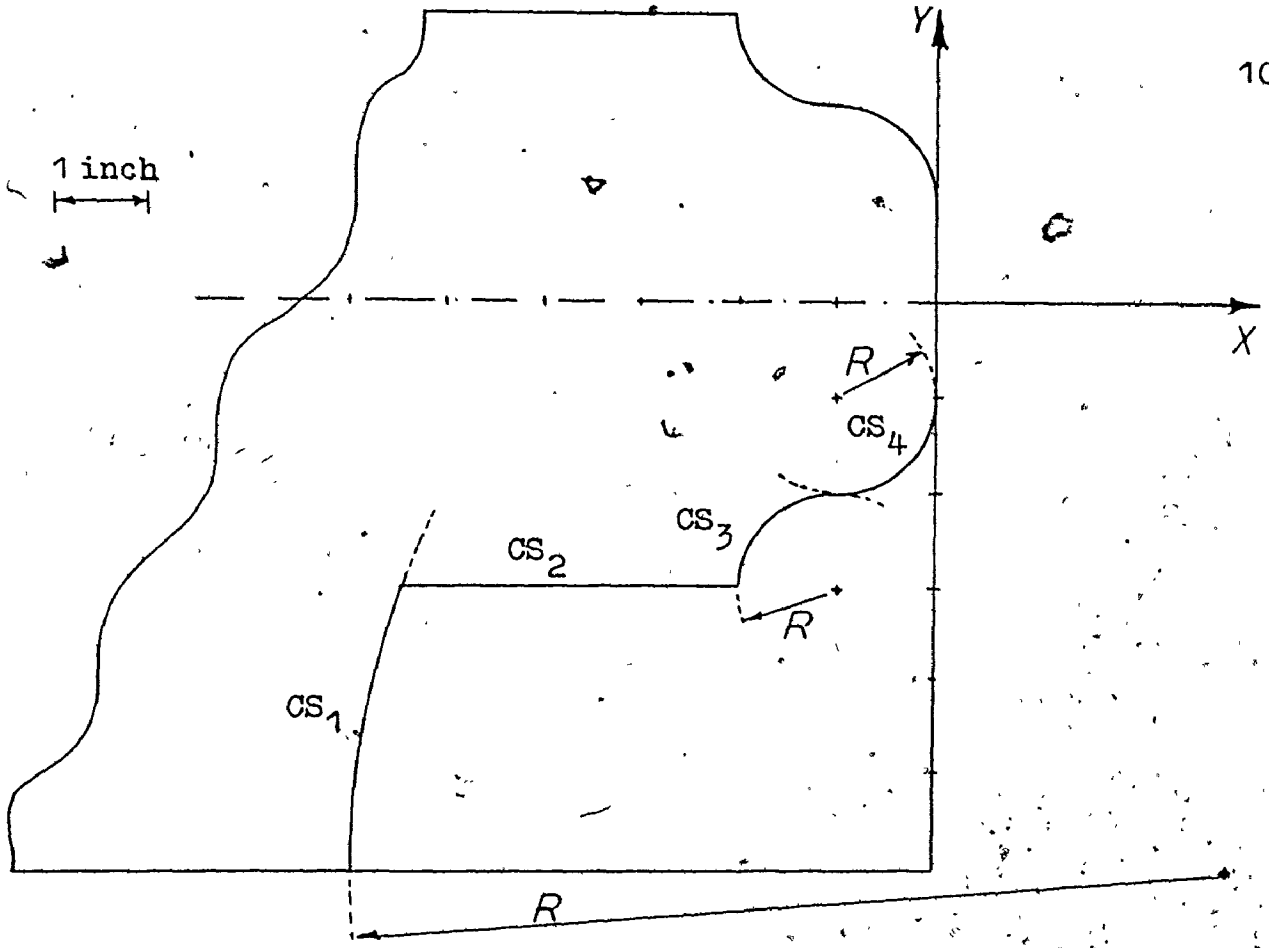

1 inch
Y
X
R
CS4
CS3
CS2
R
CS1
R

Fig. 4.4.

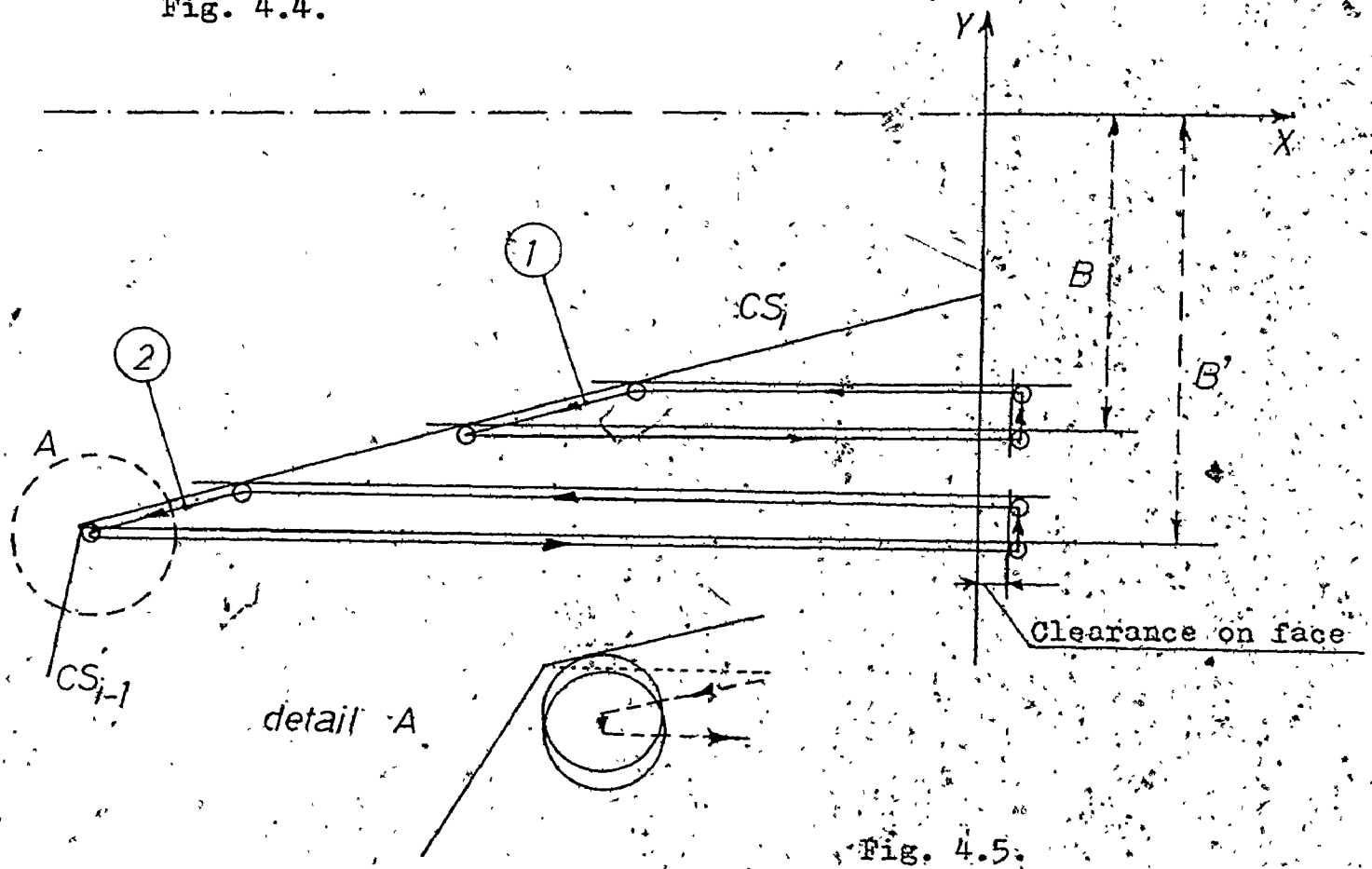

Y
X
1
2
CSi
B
B'
A
CSi-1
detail A
Clearance on face

Fig. 4.5.

```
PARTNO   ROUND SHAFT NO 2
MACHIN/LATHE,1,23 ,1
$$
$$
$$
          FISERV/3 , , ,FC,11, ,1, ,1.
$$
$$
$$
$$
PRINT/
$$  TOLERANTIES   THE LATHE
          INTOL/.   5
          CUTTOL/.  5
$$  COOLANT IS ON
          COOLNT/ON
$$  PRINT THE CUTTER LOCATIONS
          CLPRNT
$$  INSERT THE GEOMETRICAL STATEMENTS
SP      = POINT/1,-7
LX      = LINE / XAXIS
LY      = LINE / YAXIS
          CS(1)=CIRCLE/C  THR,(POINT/3,-6, ),TANTO,S
          (LINE/PARLEL,LY,X   LL, )
          CS(2)=L    /PARLEL,LX,YS ALL,3
          CS(3)=CIRCL /CENTE ,(POINT/-1,-3, ),RADIUS,1
          CS(4)=CIRCLE/CENTER,(POINT/-1,-1, ),RADIUS,1
$$  RD(I) IS THE RADIUS AT INTERSECTION OF SURFACE I-1 AND SURFACE I
          RD(1)=6
          RD(2)=3
          RD(3)=3
          RD(4)=2
          RD(5)=1
$$  KC(I) IS RELATED TO THE KIND OF SURFACE
$$        CONCAVE CIRCLE -1
$$        LINE          0
$$        CONVEX CIRCLE  1
          KC(1)=1
          KC(2)=0
          KC(3)=1
          KC(4)=-1
$$  CALL UP THE ROUGHING MACRO
          CALL/ROUGH,K=4,MDP=,2 ,FIN=    ,CL=    ,VEL=25 , AD= ,FD=
PRINT/
          PRINT/3,ALL
          END
          FINI
BEGIN SECTION    2
BEGIN SECTION    3
```

Fig. 4.6.

Part program example of turning plotted in Fig. 4.8.

Fig. 4.7. Hard copy of section IV output (part 1)

MACHINE NUMBER 1 DATE 11/30/77 PAGE 2

---- AMERICAN HUSTLER LATHE WITH GE 7542 CONTROL ----

FOUND START NO 2

WE START MACHINING A NEW SECTION

**** CANADIAN INSTITUTE OF METALWORKING - NC MACHINING DIVISION ****
GENERALIZED LATHE POST PROCESSOR

Fig. 4.7. (part 2)

```
---- AMERICAN HUSTLER LATHE WITH GE 7542 CONTROL ----
      ::::::   FOUND SHAFT NO 2
                                                                    ABX          ABZ
N094 Z-01. F006. S139                                             2.4000      -1.0000
N095 Z-01.0006                                                    2.4000      -1.0006
N096 G02 X02.6 Z-01.9171 I00.6 K00.0006                           2.6000      -1.9171
N097 G01 Z00. F000.                                               2.6000       0.0000
N098 X02.4                                                        2.4000       0.0000
N099 X02.2                                                        2.2000       0.0000
N100 Z-01. F00A. S143                                             2.2000      -1.0000
N101 Z-01.0008                                                    2.2000      -1.0008
N102 G02 X02.4 Z-01.0006 I00.6 K00.0006                           2.4000      -1.0006
N103 G01 Z00. F000.                                               2.4000       0.0000
N104 X02.2                                                        2.2000       0.0000
N105 X02.                                                         2.0000       0.0000
N106 Z-01. F00A. S147                                             2.0000      -1.0000
N107 G02 X02.2 Z-01.0006 I00. K00.                                2.2000      -1.0006
N108 G01 Z00. F000.                                               2.2000       0.0000
N109 X02.                                                         2.0000       0.0000
      ::::::   WE START MACHINING A NEW SECTION
N110 X01.6                                                        1.6000       0.0000
N111 Z-00.3992 F006. S152                                         1.6000      -.3992
N112 G03 X02.0006 Z-01. I00.8 K00.6008                            2.0006      -1.0000
N113 G01 X02.                                                     2.0000      -1.0000
N114 Z00. F000.                                                   2.0000       0.0000
N115 X01.8                                                        1.8000       0.0000
N116 X01.6                                                        1.6000       0.0000
N117 Z-00.1994 F006. S153                                         1.6000      -.1994
N118 G03 X01.6 Z-00.3992 I00.6 K00.8006                           1.6000      -.3992
N119 G01 Z00. F000.                                               1.6000       0.0000
N120 X01.6                                                        1.5000       0.0000
N121 X01.4                                                        1.4000       0.0000
N122 Z-00.0829 F00A. S163                                         1.4000      -.0829
N123 G03 X01.6 Z-00.1994 I00.4 K00.9171                           1.6000      -.1994
N124 G01 Z00. F000.                                               1.6000       0.0000
N125 X01.4                                                        1.4000       0.0000
N126 X01.2                                                        1.2000       0.0000
N127 Z-00.0197 F00A. S173                                         1.2000      -.0197
N128 G03 X01.4 Z-00.0829 I00.2 K00.9803                           1.4000      -.0829
N129 G01 Z00. F000.                                               1.4000       0.0000
N130 X01.2                                                        1.2000       0.0000
N131 X01.                                                         1.0000       0.0000
N132 Z-00.0005 F006. S134                                         1.0000      -.0005
N133 G02 X01.2 Z-01.9803 I00. K01.0005                            1.2000      -1.9803
N134 G01 Z00. F000.                                               1.2000       0.0000
N135 X07. Z01.                                                    7.0000       1.0000
N136 G04 X02.5 T000 M02                                           7.0000       1.0000
      ACCUM. CUT TIME   0.77   MIN.          ACCUM. DWELL TIME   0.00 MIN.
```

**** CANADIAN INSTITUTE OF MECHANICAL WORKING PROCESSOR (ACHIEVED POST PROCESSOR REVISION) ****

Fig. 4.7. (part 3)

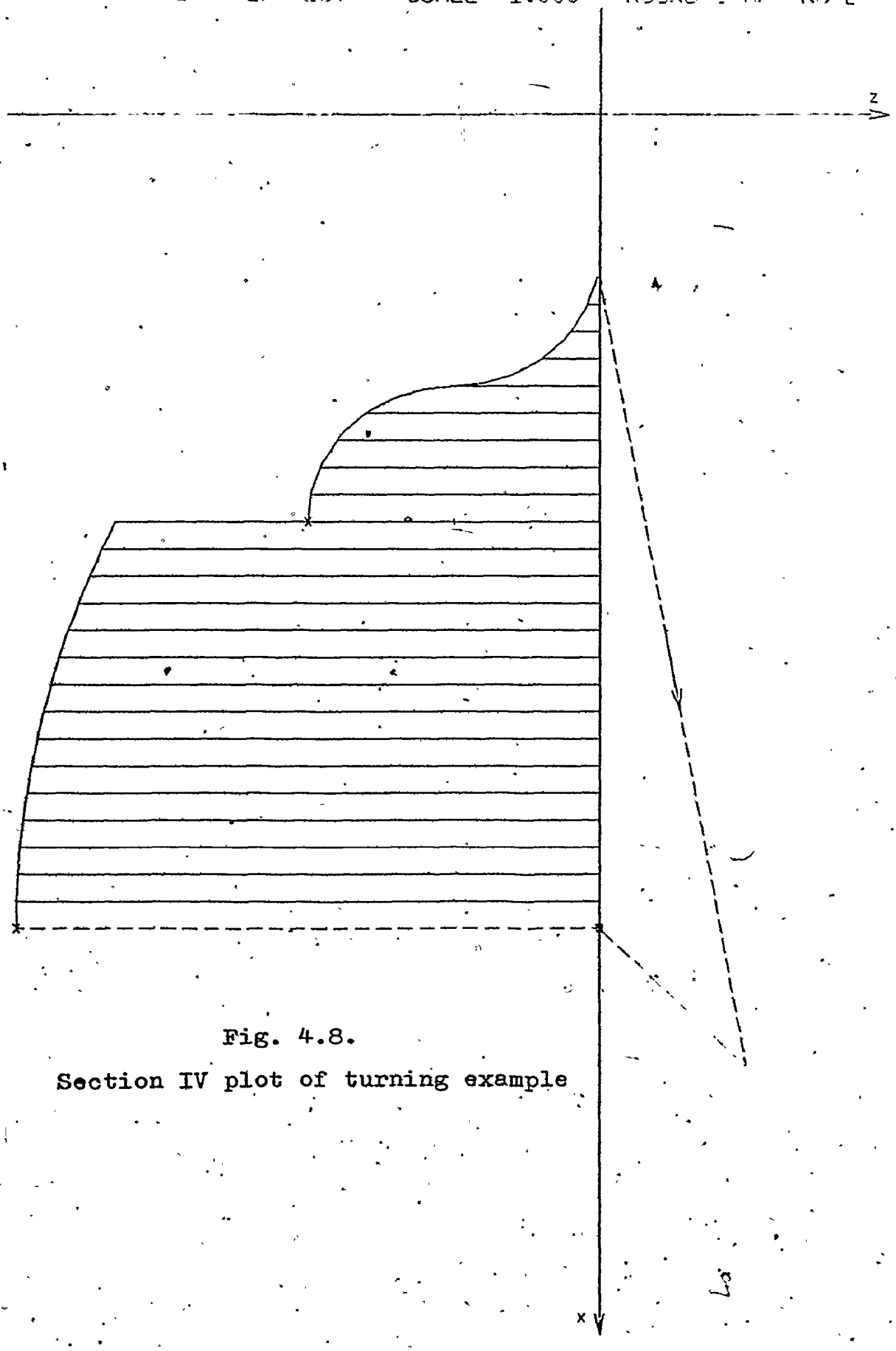


Fig. 4.8.

Section IV plot of turning example

Example 3.

A complicated milling part such as the one depicted in Fig. 4.9. can only be handled by the programming sets RGH1 and RGH2 (not by the SPIRAL set). This configuration consists of 7 boundary surfaces. Because of the abundance of CL data the printing of the section III output was omitted in this work. Fig. 4.10. and Fig. 4.11. respectively represent the section IV plot and the section IV hard copy output.

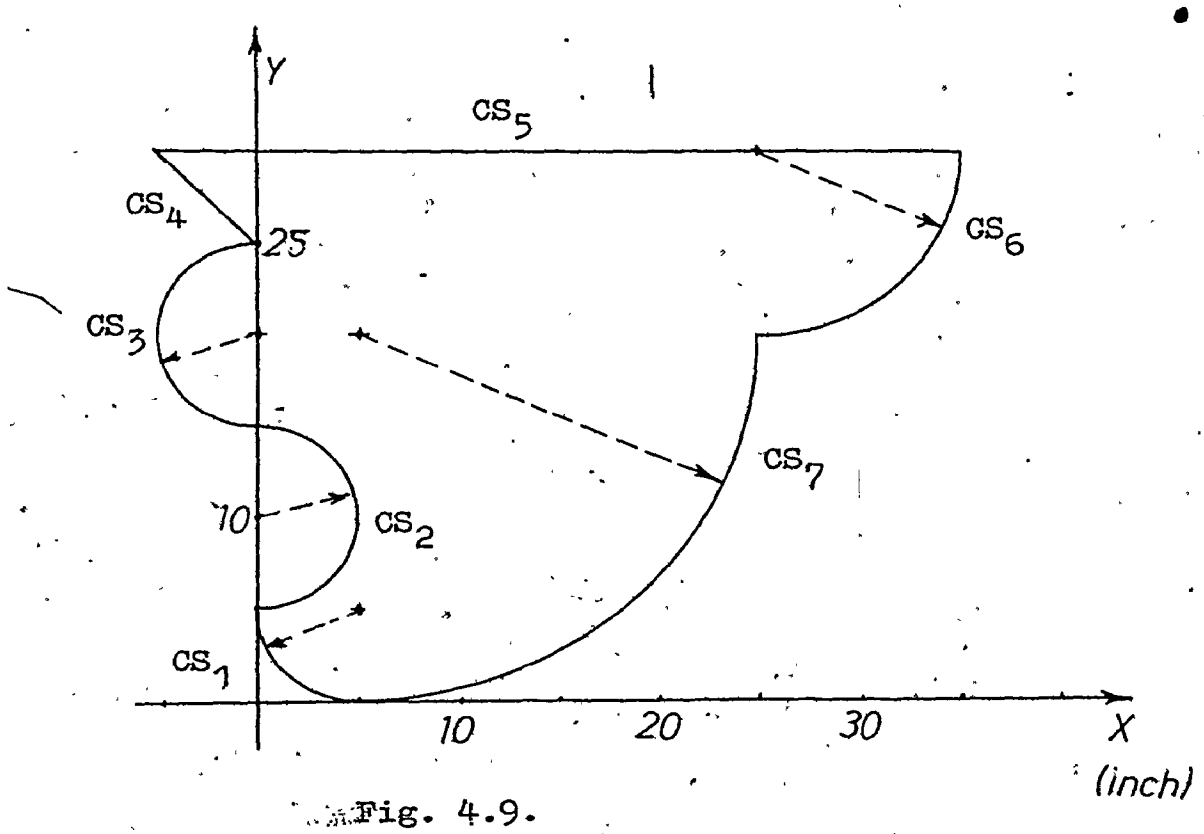
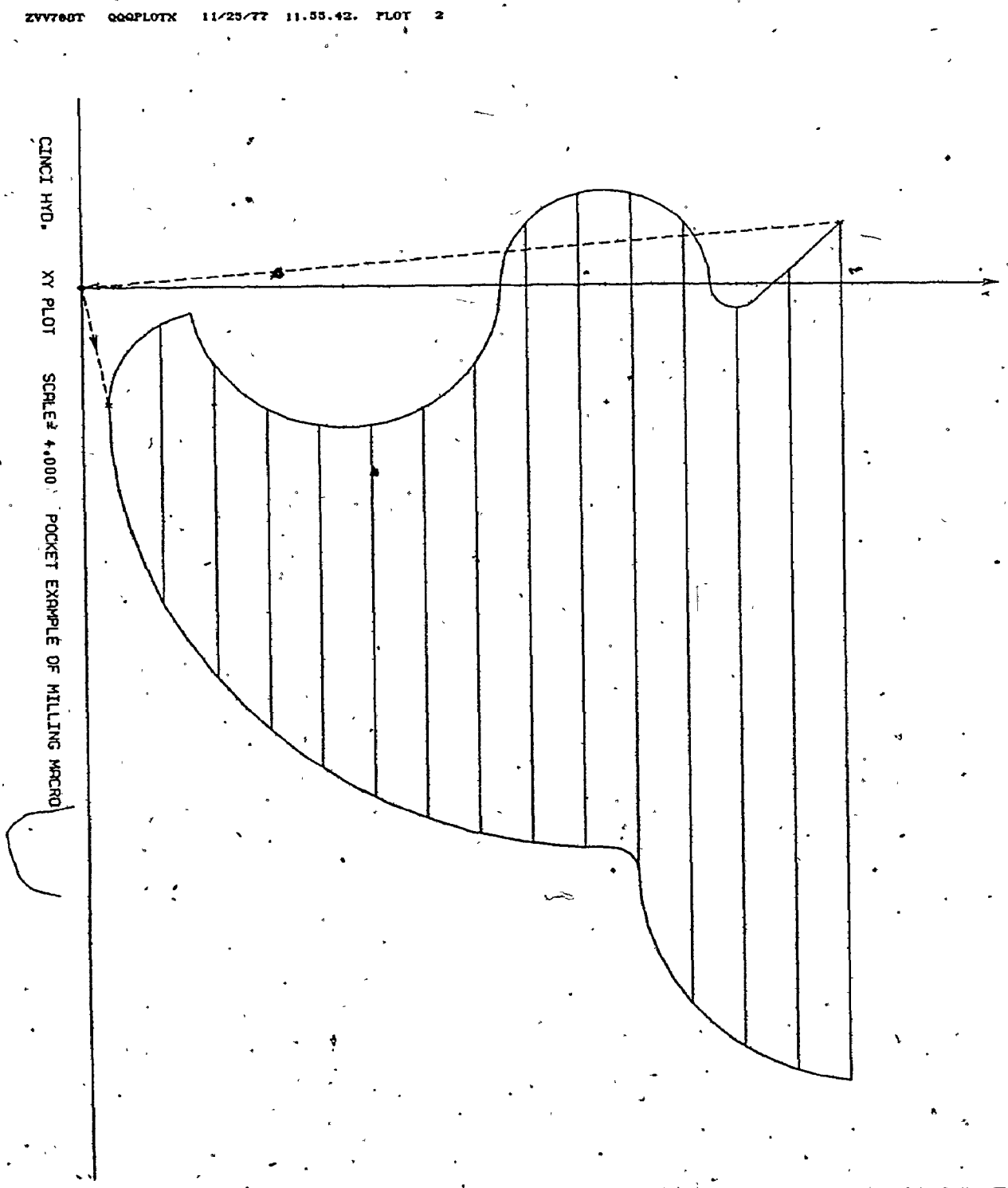


Fig. 4.9.

Fig. 4.10.A. Section IV plot describing toolpath of pocket example in X - Y plane.



CANADIAN INSTITUTE OF METALWORKING
11/25/77

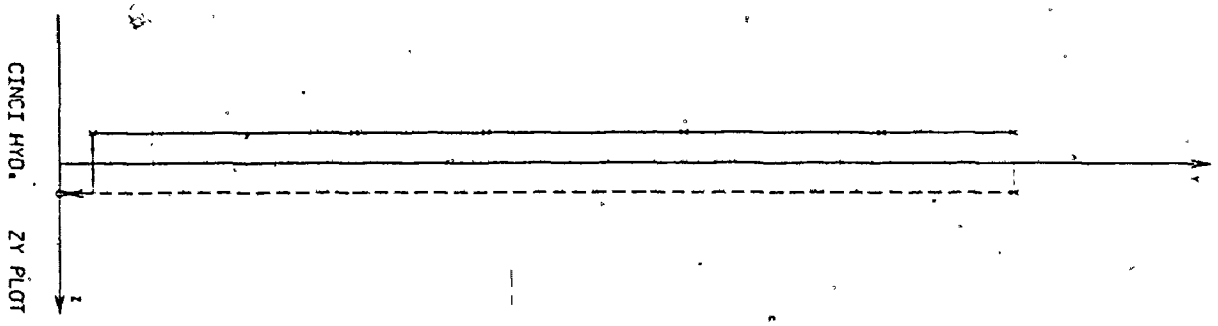
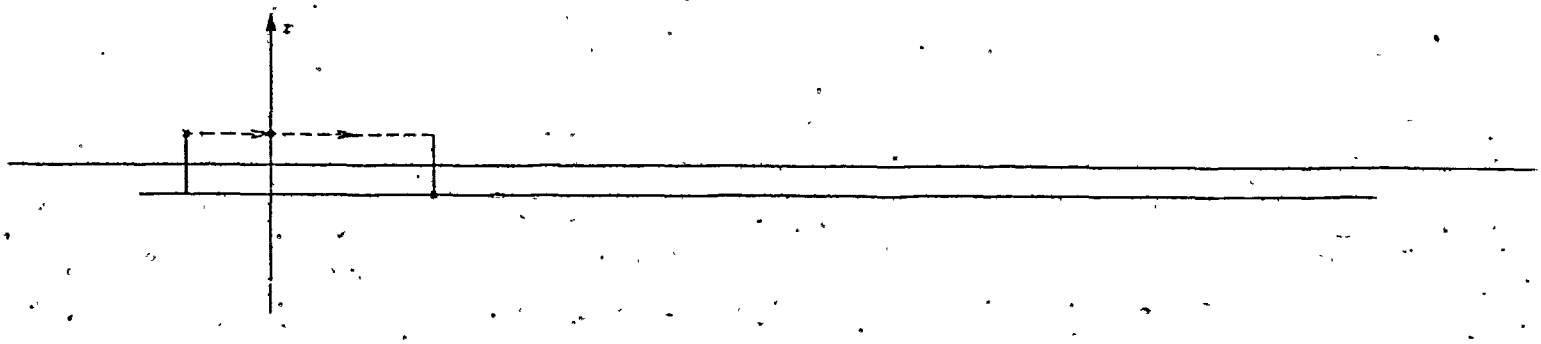


Fig. 4.10.B.
Section IV plot describing toolpath
of pocket example in Y - Z plane
in X - Z plane

ZVVT88T GGGPLOTX 11/25/77 11.55.42. PLOT 8



```
                 C I N A C 1  300  /  A P T  III  P O S T  P R O C E S S O R
          A C R A M A T I C   I V   3   A X E S   C O N T O U R I N G   C O N T R O L

 H/N    G      P         X        C        Y          R        Z        F      S     M   SCLTAPE RPM
           SAMPLE PART PROGRAM FOR POCKETING
LEADER/    72.0
 H001   G01         X       0          Y        0          Z  10000  F 200        MC5$     53
 N001               X   43193          Y     9639                    F1000        M13$     64
 N002               X   50000          Y    10000                    F 377           $     64
 N003                                                      Z- 9800   F  10           $     67
 N004                                                      Z- 10000  F   2           $     67
 N005               X   50004                                                        $     77
           ****************************************************************

           BEGINNING OF SEMIROUGHING CUT

           ****************************************************************
 N006   G02         X   50000          Y    50000                                    $     94
 N007               X   11088          Y    40730                    F 200        M13$     94
 N008   G01         X   11026          Y    41000                    F   6           $     94
 N009               X   11020          Y    41016                                    $     99
 N010   G03         X       0          Y   100000                                    $    107
 N011               X       0          Y   160004                    F 200        M13$    107
 N012   G02         X       0          Y   200000                                    $    111
 N013               X       0          Y   239994                    F 200        M13$    111
 N014   G03         X       0          Y   250000                                    $    115
 N015               X    7074          Y   257074                    F 200        M13$    115
 N016   G01         X   25858          Y   290000                                    $    115
 N017               X  333166                                                        $    126
 N018               X  339443                                        F   2           $    125
 N019               X  339438                                                        $    133
 N020   G02         X  250000          Y   300000                                    $    140
 N021               X  250000          Y   210006                    F 200        M13$    140
 N022   G03         X  250000          Y   200000                                    $    143
 N023               X  239994          Y   200000                    F 200        M13$    143
 N024   G02         X   50000          Y   200000                                    $    148
 N025               X   51276          Y    10006                    F 200        M13$    148
 N026   G01         X   50000                                        F   1           $    148
           ****************************************************************

           END OF THE SEMIROUGHING CUT

           ****************************************************************
 N027                                  Y    10004                                    $    165
 N028   G03         X   50000          Y   200000                                    $    173
 N029               X  134850          Y    30004                    F 200        M13$    173
 N030   G01         X   15364          Y    30000                                    $    173
 N031   G02         X   50000          Y    50000                                    $    185
 N032               X   11026          Y    41018                    F 200        M13$    185
 N033   G03         X       0          Y   100000                                    $    190
 N034               X   33174          Y    50002                    F 200        M13$    190

 MACHINING TIME     9.854 MINUTES                         TAPE LENGTH    5.74 FEET
```

Fig. 4.11. (part 1)

C I N A C 1 300 / A P T III P O S T P R O C E S S O R

SAMPLE PART PROGRAM FOR POCKETING

H/N	G	P	X	Y	R	Z	F	S	M	SC-TAPE RPM
N035	G01		X 166612	Y 50000						195
N036	G03		X 50000	Y 200000						204
N037			X 188560	Y 70302			F 200		M13	204
N038	G01		X 51966	Y 70000						204
N039	G03		X 0	Y 100000						211
N040			X 53164	Y 90000			F 200		M13	211
N041	G01		X 204914							216
N042	G03		X 50000	Y 200000						225
N043			X 217328	Y 110002			F 200		M13	225
N044	G01		X 59166	Y 110000						225
N045	G03		X 0	Y 100000						232
N046			X 51966	Y 130000			F 200		M13	232
N047	G01		X 226630							237
N048	G03		X 50000	Y 200000						246
N049			X 233298	Y 150002			F 200		M13	246
N050	G01		X 33174	Y 150000						246
N051	G03		X 0	Y 100000						253
N052			X 0	Y 160004			F 200		M13	253
N053	G02		X 0	Y 200000						257
N054			X- 26454	Y 170004			F 200		M13	257
N055	G01		X 237612	Y 170000						262
N056	G03		X 50000	Y 200000						270
N057			X 239732	Y 190000			F 200		M13	270
N058	G01		X- 38724							275
N059	G02		X 0	Y 200000						284
N060			X- 38724	Y 209998			F 200		M13	284
N061	G01		X 250000	Y 210000						284
N062	G03		X 250000	Y 300000						305
N063			X 316572	Y 231004			F 200		M13	305
N064	G01		X- 26450	Y 230000						310
N065	G02		X 0	Y 200000						317
N066			X 0	Y 239994			F 200		M13	317
N067	G03		X 0	Y 250000						321
N068			X 11004	Y 250000			F 200		M13	321
N069	G01		X 324828							325
N070	G03		X 250000	Y 300000						335
N071			X 334848	Y 270002			F 200		M13	335
N072	G01		X- 5858	Y 270000						335
N073			X- 25858	Y 290000						338
N074						Z 10000	F 10			341
N075									M09	345
N076									M05	347
N077			X- 164	Y 1844			F1000			351
N078			X 0	Y 0			F 371			351
N079									M02	353

LEADER/ 72.0

SAMPLE PART PROGRAM FOR POCKETING

MACHINING TIME 27.874 MINUTES

TAPE LENGTH 13.16 FEET

END OF PART PROGRAM

Fig. 4.11. (part 2)

REFERENCES

1) Williams, J.C., "NC Comes Full Circle in the U.S.Army", Proc. of NC Society, De Vries, Anaheim Ca., 1971.

2) Koloc, J., "Some Practical Aspects of Computer-Aided Programming via Time-Sharing Network", CIRP ANNALS, 1975.

3) McCarroll, J.D., "Computer-Aided Part Programming for Numerical Control, an Industry Study", Institute of Science and Technology, University of Michigan, Ann Arbor, 1969.

4) Control Data Corporation, "APT Reference Manual 6000, Version 2", Sunnyvale, Ca., 1971.

5) IBM, "System/360 APT NC Processor (360A-CN-10X) Version 4 Part Programming Manual", IBM, White Plains, NY., 1972.

6) Koloc, J., "Some Practical Aspects of Reliability of Computer-Aided Programming of NC Machine Tools", CIRP ANNALS, 1976.

7) Dunsford, G.C., "Numerical Control Programming Language for Lathes", M. Eng. Thesis, McMaster University, 1972.

8) IIT Research Institute, "APT Part Programming", McGraw-Hill, 1967.

9) EXAPT 1, "Part Programmer Reference Manual", Aachen, Germany, 1967.

10) EXAPT Annual General Meeting June 13th, 1975, "Minutes", Aachen, Germany, 1975.

11) EXAPT Annual General Meeting May 14th, 1976, "Minutes", Aachen, Germany, 1976.

12) Oudolf, T.H., "Development of a Programming System for NC Machining Centers", CIRP ANNALS, 1976.

13) Grotheer, W., "Introduction and Application of the EXAPT System at Demag Metallgewinning", EXAPT Annual General Meeting June 13th, 1975, "Minutes", Aachen, Germany, 1975.

14) Okino, N., Kakazu, Y., and Kubo, H., "TIPS 1 : Technical Information Processing System for Computer-Aided Design,

Drawing and Manufacturing", Proc. of 2nd IFIP/IFAC Int. Conf., PROLAMAT, Budapest, 1973.

15) Braid, I.C. and Lang, C.A., "Computer-Aided Design of Mechanical Components with Volume Building Bricks", Proc. of 2nd IFIP/IFAC Int. Conf., PROLAMAT, Budapest, 1973.

16) Yellowley, I., "The Development of Machinability Testing Methods, with Specific Reference to High Strength Thermal Resistant Work Materials", Ph.D.Thesis, Victoria University of Manchester, 1974.

17) Yellowley, I. and Barrow, G., "The Influence of Thermal Cycling on Tool Life in Peripheral Milling", Metal Cutting Seminar, 16th International M.T.D.R. Conference, Manchester, 1975. Published in International Journal of Machine Tool Design Research, Vol. 16, p. 1-12, 1976.

18) Tlusty, J., Yellowley, I. and Konrad, G., "Tool Wear in the Peripheral Milling of a Low Carbon Steel", ASME publication 76-WA/Prod-35, 1976.

19) Yellowley, I., "The Economics of the Peripheral Milling Process", Metalworking Research Group, Report no. 82, McMaster University, 1976.

20) Yellowley, I., "The Selection of Cutting Conditions in Peripheral Milling", to be published.

21) Yellowley, I., Wong, A. and De Smit, B., "The Selection of Cutter Diameter in Peripheral Milling", to be published.

22) Siddall, J.N., "Analytic Decision-Making in Engineering Design", Prentice-Hall, Englewood Cliffs, N.J., 1972.

23) Siddall, J.N., "OPTISEP, Designers Optimization Subroutines", User's Manual, McMaster University, Hamilton, Canada, 1975.

24) Digital Equipment Corporation, "BASIC/RT11 Language Reference Manual", Digital, 1974.

25) Colman, H.L., "APT Today and Tomorrow", Institute of Science and Technology, University of Michigan, Ann Arbor, 1967.

26) Gouldson, C., "Computer Part Programming and Tooling Guide

for American Hustler Lathe GE 7542 Control", C.I.M., Hamilton, 1975.

27) Leslie, W.H.P., "Numerical Control User's Handbook", Mc Graw Hill, 1970.

28) French, D., "APT Macros for the Production of Drop-Forging Dies", University of Waterloo, Ontario, 1973.

29) Wong, A. and Yellowley, I., "The Use of a Low Cost Minicomputer System in the Preparation of Part Programs for Numerically Controlled Machine Tools", Metalworking Research Group, Report no. 72, McMaster University, 1976.

30) Digital Equipment Corporation, "PDP 11/34 Processor Handbook", DEC, 1976.

31) Hatvany, J., "Interactive, Conversational and Graphic Programming", Proc. of 2nd IFIP/IFAC Int. Conf., PROLAMAT, Budapest, 1973.

32) Cremerius, J., "APT/IGS : State of the Art in NC Graphics", Proc. of 2nd IFIP/IFAC Int. Conf., PROLAMAT, Budapest, 1973.

33) El Maraghy, H., "The Effects of Computer Graphics in Design and Manufacturing", Metalworking Research and Design Group, Report no. 104, McMaster University, June 1977.

34) Crestin, J.P. and Paillard, J.P., "A Small Graphical System for Programming NC Machines", Proc. of 2nd IFIP/IFAC Int. Conf., PROLAMAT, Budapest, 1973.

35) NC Sciences, Inc., "Practical APT Part Programming", Philadelphia, Pa., USA.

36) EXAPT 2, "Part Programmer Reference Manual", Aachen, Germany

37) Engelskirchen, W.H., "NC in Europe", Proc. of NC Society, De Vries, Anaheim, Ca., 1971.

38) Macwrek, I. and Vencovsky, J., "NC Programming in C.K.D. Praha", Proc. of 2nd IFIP/IFAC Int. Conf., PROLAMAT, Budapest, 1973.

39) Simon, W., "The Numerical Control of Machine Tools", Edward Arnold, London, England, 1970.

40) Bjorke, O., "Software Production - The Bottleneck of Future Manufactur ng Systems", CIRP ANNALS, 1975.

41) Private Contact with C.J.Gouldson, Canadian Institute of Metalworking, Hamilton.

42) Private Contact with J.E.Grozier, Canadian Institute of Metalworking, Hamilton.

APPENDICES.

APPENDIX 1 : Flowcharts.

1. APT system macro architecture.

   1.1. The milling macros.

      1.1.1. General structure of the milling programs.

      1.1.2. Common preparatory macros for the milling macros RGH1, RGH2 and SPIRAL.

      1.1.3. The main pocketing macros.

         1.1.3.1. Flowchart FINMAC.
         1.1.3.2. Flowchart RGH1.
         1.1.3.3. Flowchart RGH2.
         1.1.3.4. Flowchart SPIRAL.

   1.2. The turning macros.

      1.2.1. Flowchart ROUGH.

      1.2.2. Flowchart SUPV1.

2. The preprocessor software package architecture.

   General lay-out.

   Flowcharts.

APPENDIX 2 : Program listings.

1. APT software

   1.1. Milling macros.

      1.1.1. Common preparatory macros for the milling macros.

      1.1.2. The main pocketing macros.

         1.1.2.1. FINMAC.
         1.1.2.2. RGH1.
         1.1.2.3. RGH2.
         1.1.2.4. SPIRAL.

   1.2. Turning macros.

      1.2.1. Barstock turning macro : ROUGH.

      1.2.2. Macro for machining forgings : SUPV1.

2. Preprocessor (minicomputer) software.

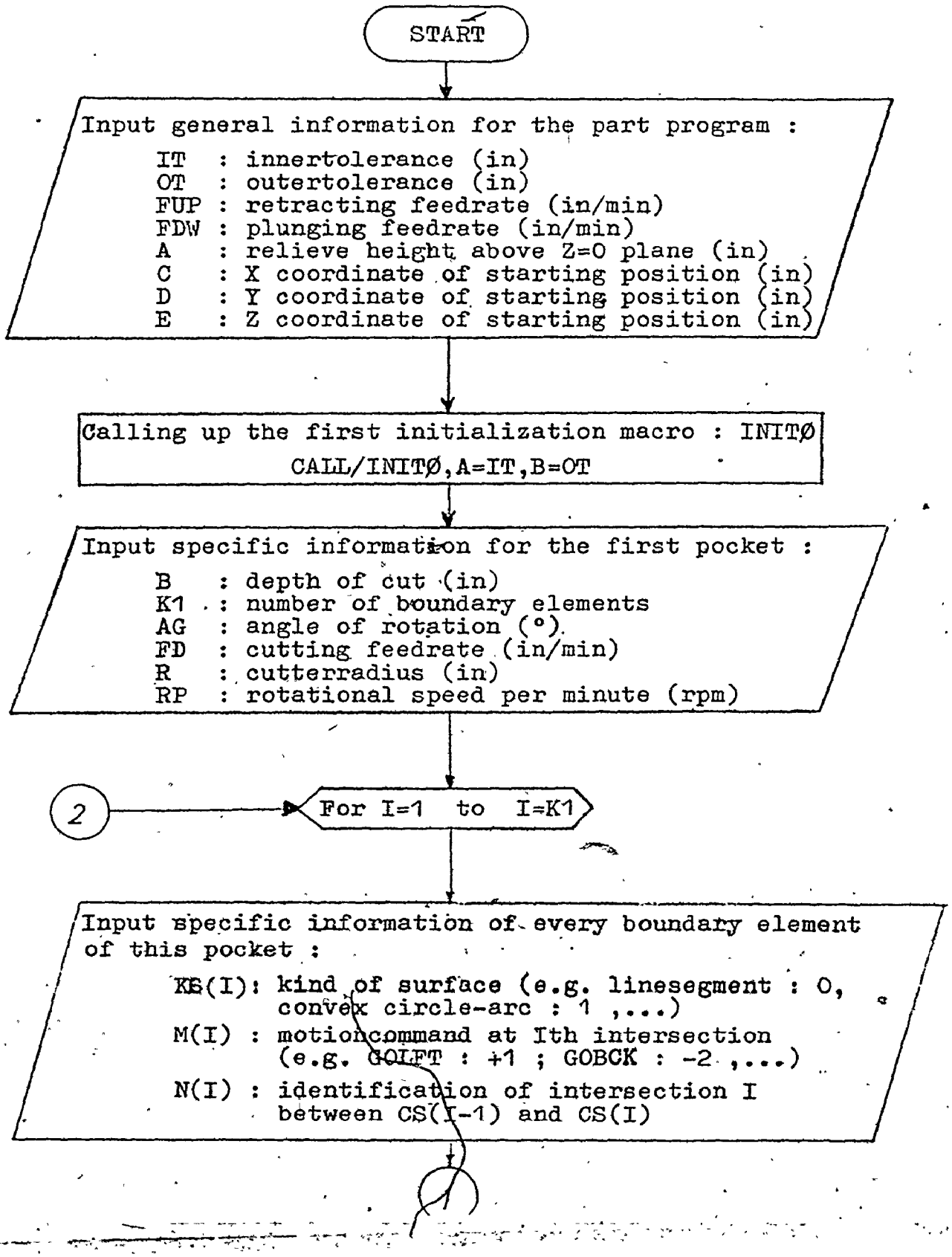
APPENDIX 3 : Auxiliary calculations concerning the SPIRAL macro.

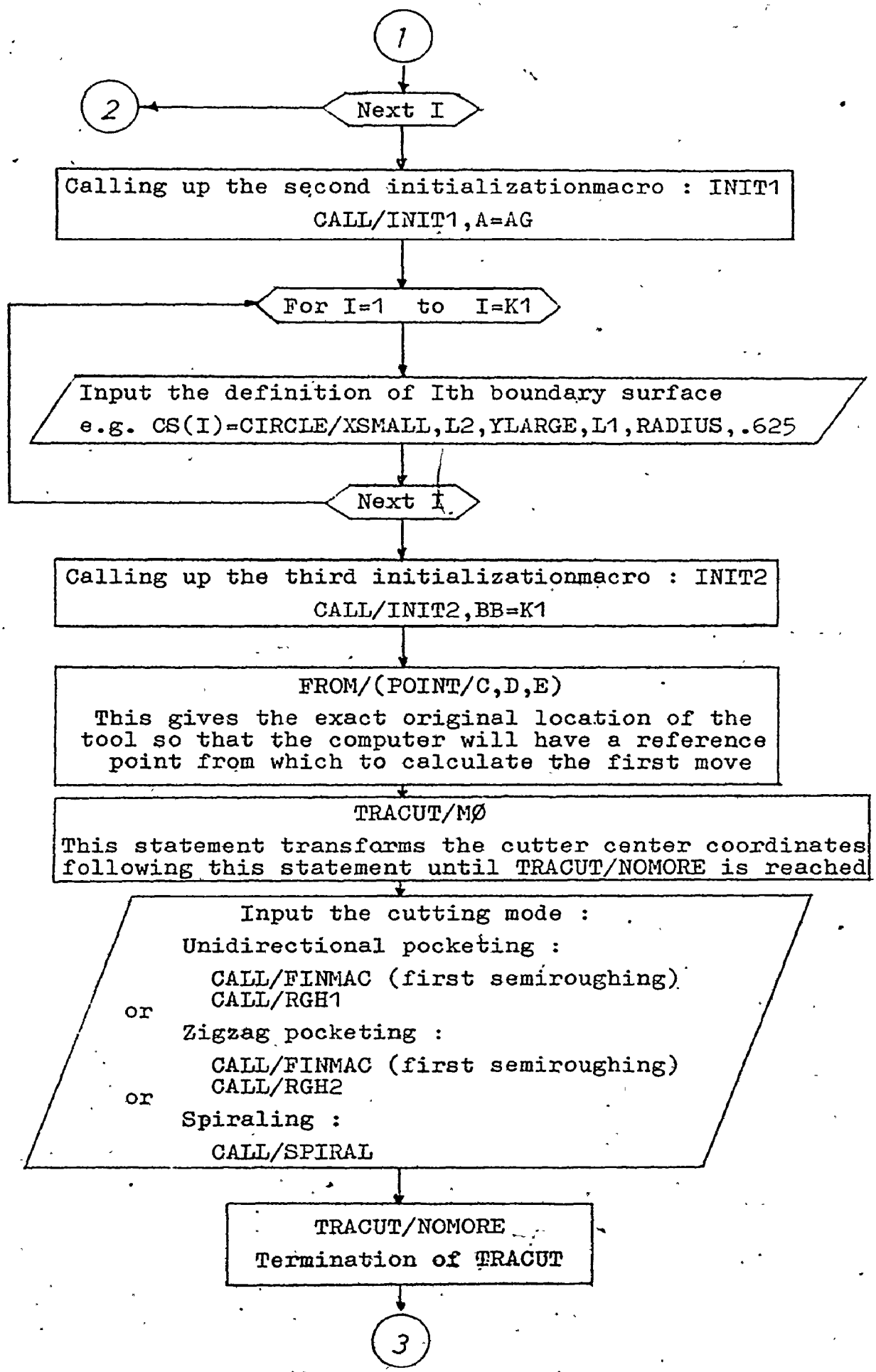
Appendix 1 : FLOWCHARTS

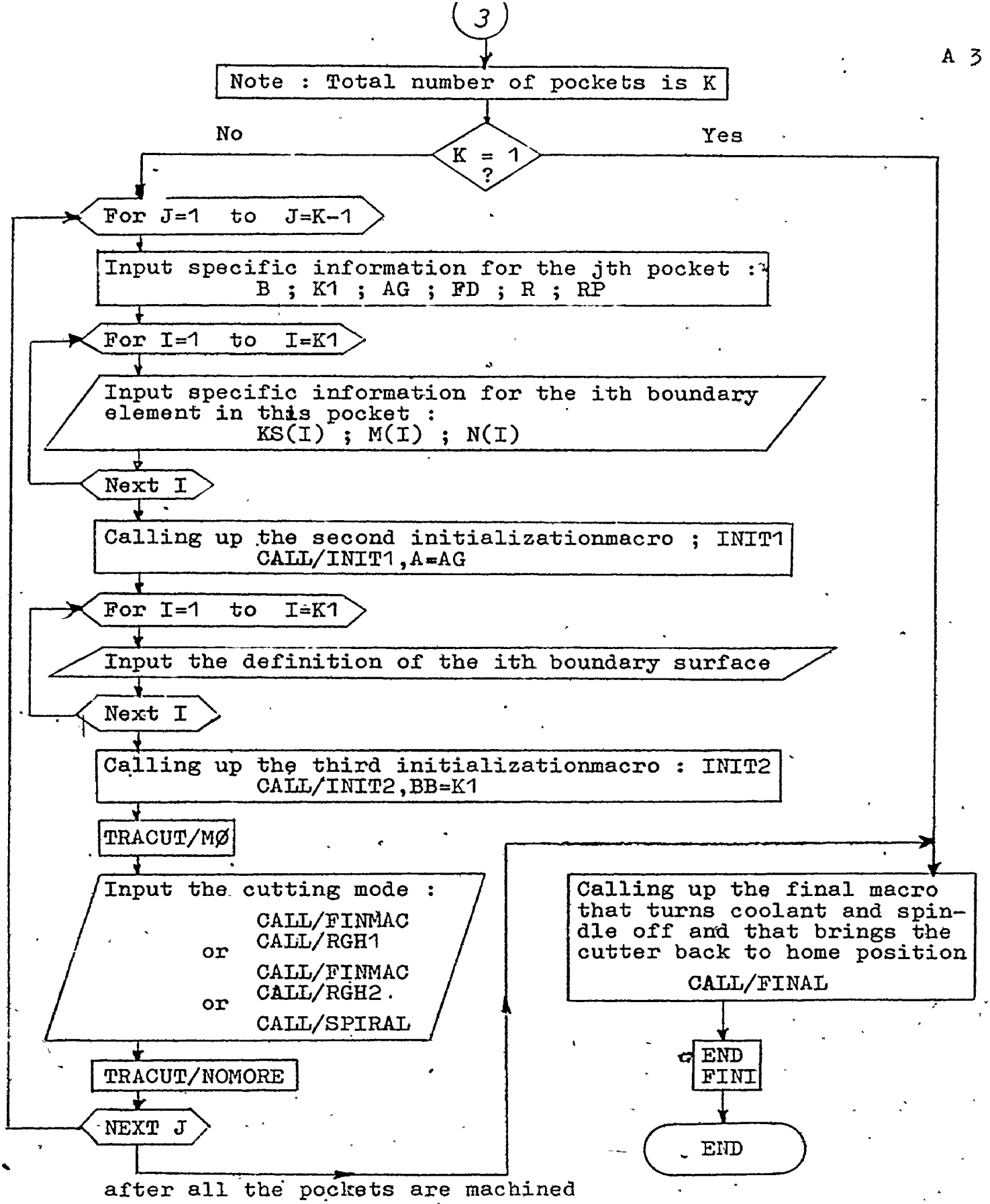
1. APT system macro architecture

1.1. The milling macros

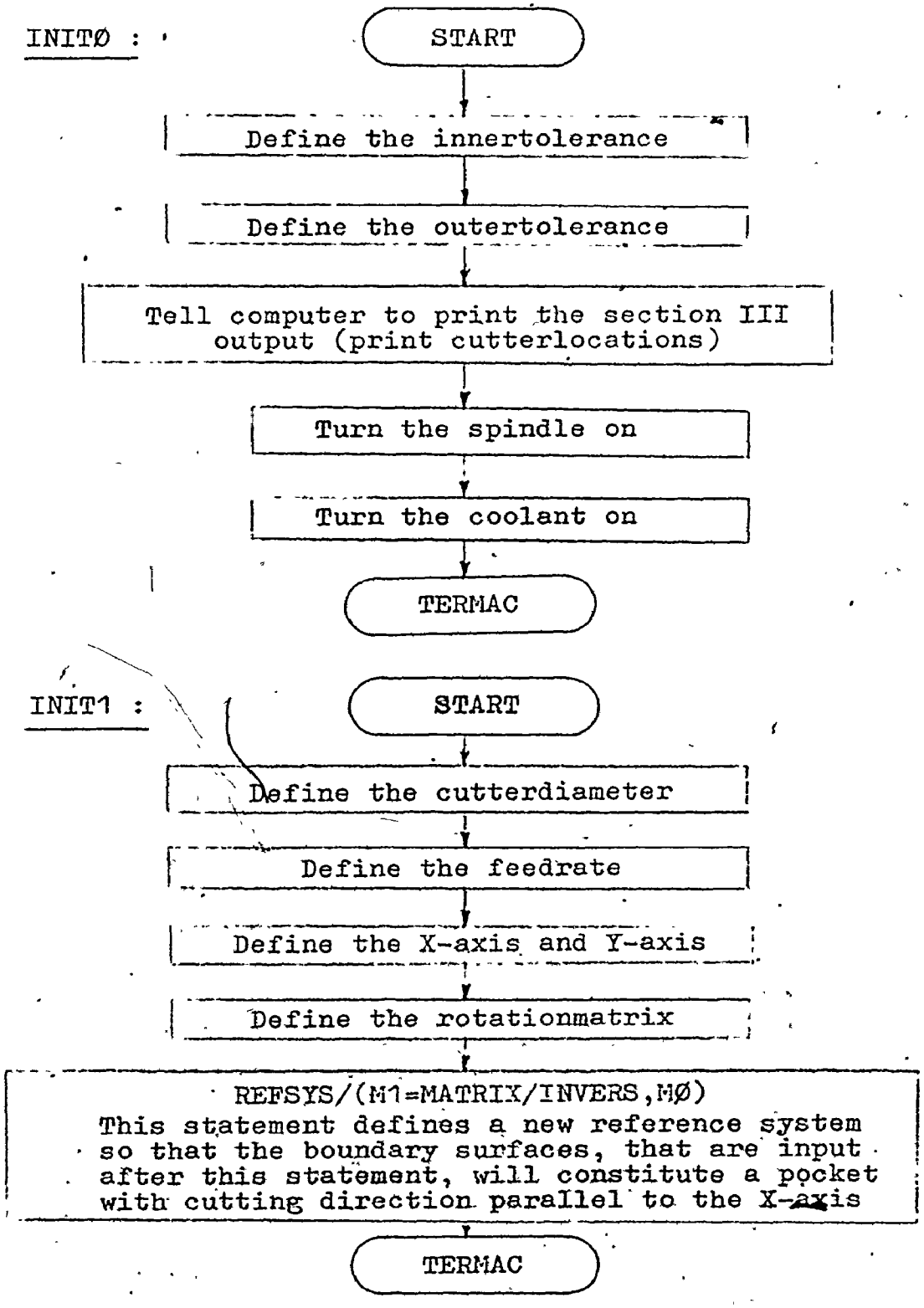
1.1.1. General structure of the milling programs



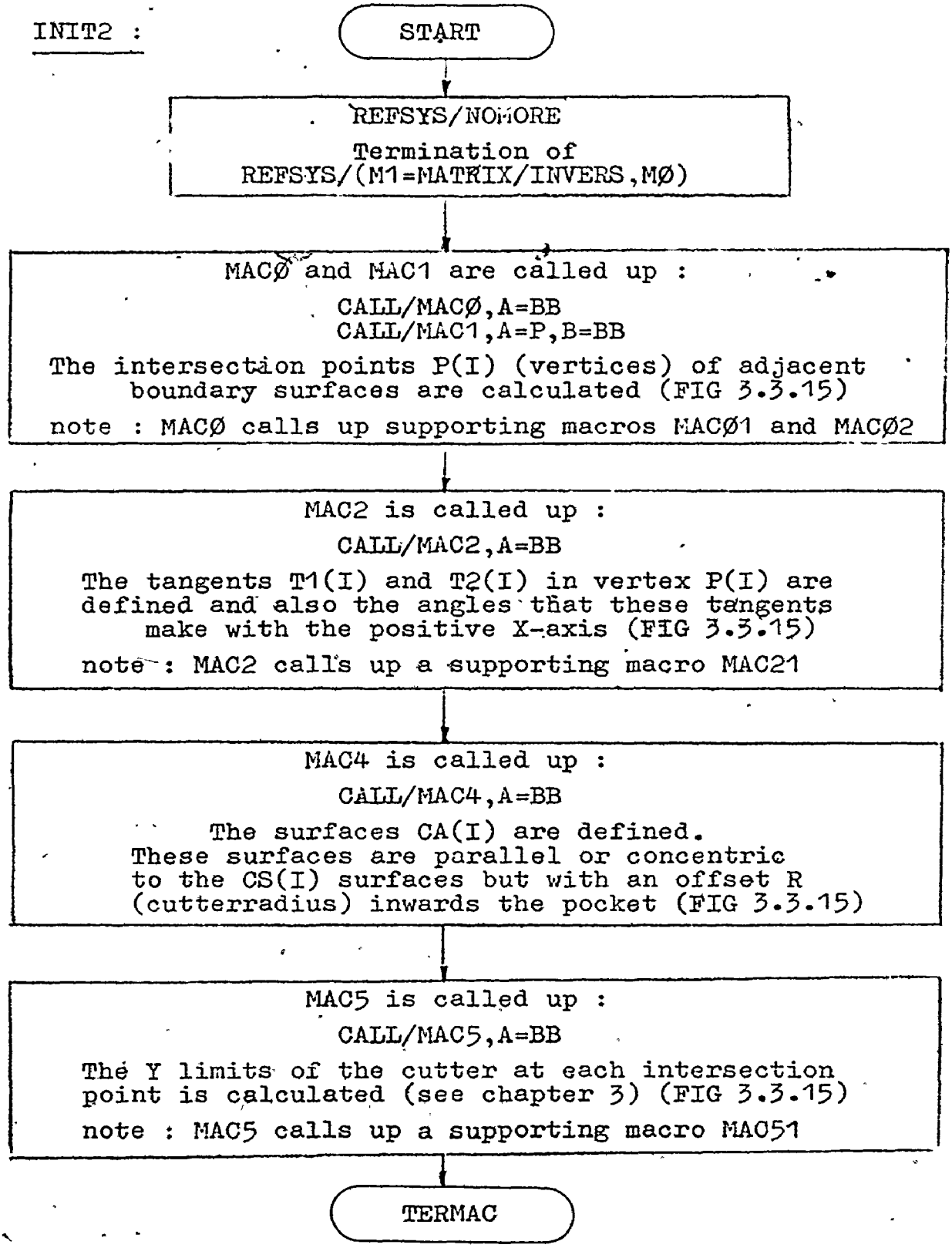

1
2
Next I
Calling up the second initializationmacro : INIT1
CALL/INIT1,A=AG
For I=1 to I=K1
Input the definition of Ith boundary surface
e.g. CS(I)=CIRCLE/XSMALL,L2,YLARGE,L1,RADIUS,.625
Next I
Calling up the third initializationmacro : INIT2
CALL/INIT2,BB=K1
FROM/(POINT/C,D,E)
This gives the exact original location of the tool so that the computer will have a reference point from which to calculate the first move
TRACUT/MØ
This statement transforms the cutter center coordinates following this statement until TRACUT/NOMORE is reached
Input the cutting mode :
Unidirectional pocketing :
CALL/FINMAC (first semiroughing)
CALL/RGH1
or
Zigzag pocketing :
CALL/FINMAC (first semiroughing)
CALL/RGH2
or
Spiraling :
CALL/SPIRAL
TRACUT/NOMORE
Termination of TRACUT
3


3
Note : Total number of pockets is K
No
K = 1 ?
Yes
For J=1 to J=K-1
Input specific information for the jth pocket :
B ; K1 ; AG ; FD ; R ; RP
For I=1 to I=K1
Input specific information for the ith boundary element in this pocket :
KS(I) ; M(I) ; N(I)
Next I
Calling up the second initializationmacro ; INIT1
CALL/INIT1,A=AG
For I=1 to I=K1
Input the definition of the ith boundary surface
Next I
Calling up the third initializationmacro : INIT2
CALL/INIT2,BB=K1
TRACUT/MØ
Input the cutting mode :
CALL/FINMAC
CALL/RGH1
or
CALL/FINMAC
CALL/RGH2.
or
CALL/SPIRAL
TRACUT/NOMORE
NEXT J
after all the pockets are machined
Calling up the final macro that turns coolant and spindle off and that brings the cutter back to home position
CALL/FINAL
END
FINI
END

1.1.2. Common preparatory macros for the milling macros RGH1, RGH2 and SPIRAL

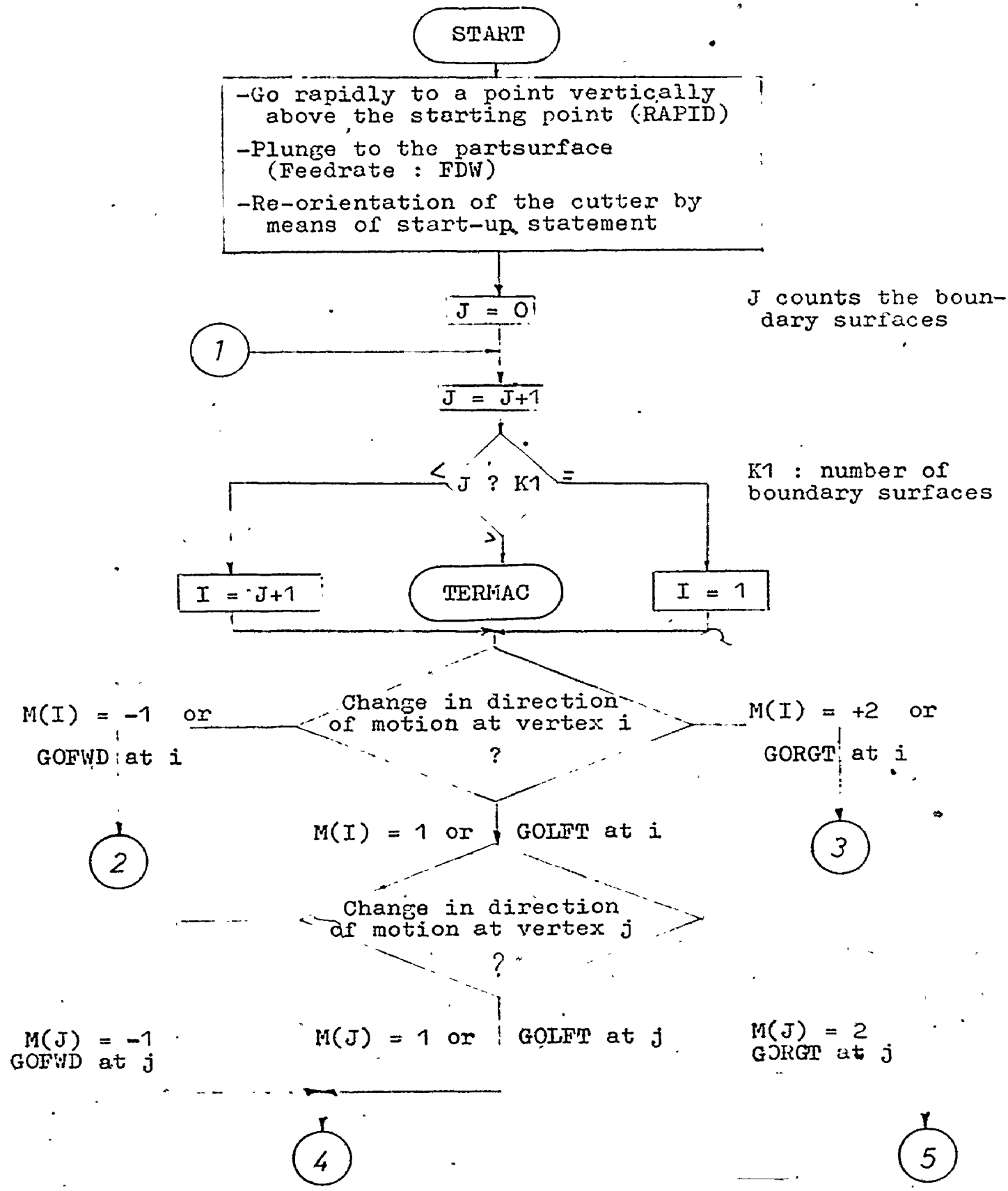


INIT2 :


START
REFSYS/NOMORE
Termination of
REFSYS/(M1=MATRIX/INVERS,MØ)
MACØ and MAC1 are called up :
CALL/MACØ,A=BB
CALL/MAC1,A=P,B=BB
The intersection points P(I) (vertices) of adjacent boundary surfaces are calculated (FIG 3.3.15)
note : MACØ calls up supporting macros MACØ1 and MACØ2
MAC2 is called up :
CALL/MAC2,A=BB
The tangents T1(I) and T2(I) in vertex P(I) are defined and also the angles that these tangents make with the positive X-axis (FIG 3.3.15)
note : MAC2 calls up a supporting macro MAC21
MAC4 is called up :
CALL/MAC4,A=BB
The surfaces CA(I) are defined.
These surfaces are parallel or concentric to the CS(I) surfaces but with an offset R (cutterradius) inwards the pocket (FIG 3.3.15)
MAC5 is called up :
CALL/MAC5,A=BB
The Y limits of the cutter at each intersection point is calculated (see chapter 3) (FIG 3.3.15)
note : MAC5 calls up a supporting macro MAC51
TERMAC

1.1.3. The main pocketing-macros.

1.1.3.1. Flowchart FINMAC (semiroughing macro) and auxiliary macros FIN1 and FIN2



Change in direction of motion at vertex j

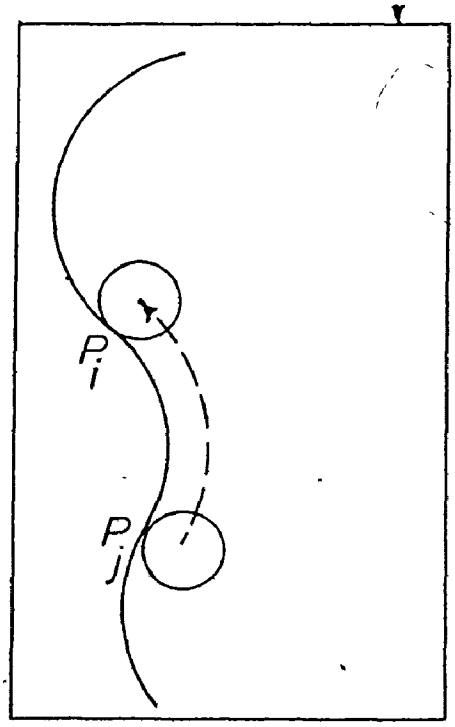
M(J) = -1
GOFWD at j

M(J) = 1
GOLFT at j

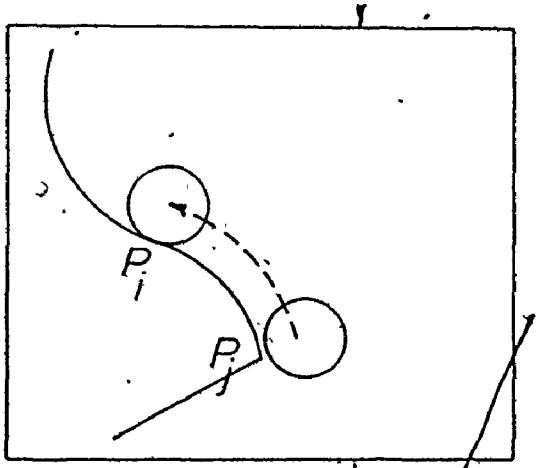
M(J) = 2
GORGT at j

TLON,GOFWD/CA(J),TANTO,CA(I)

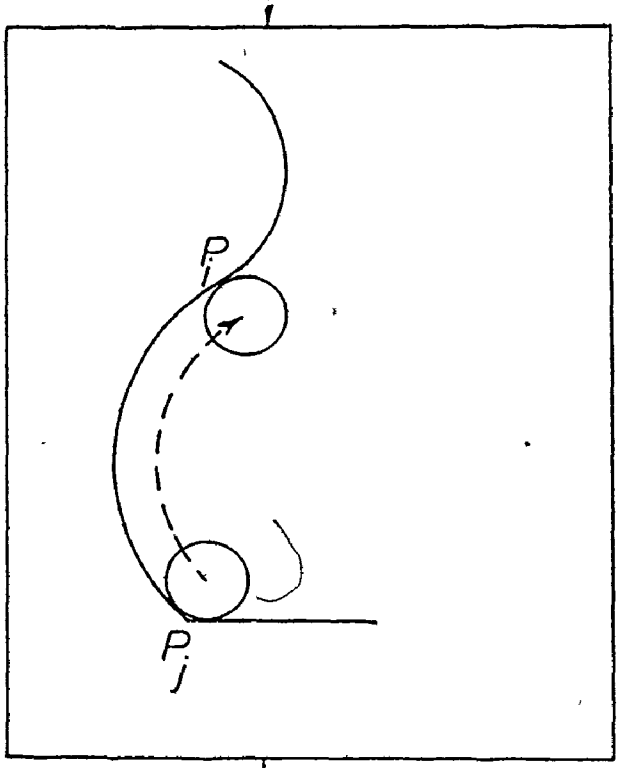
TLON,GORGT/CA(J),TANTO,CA(I)



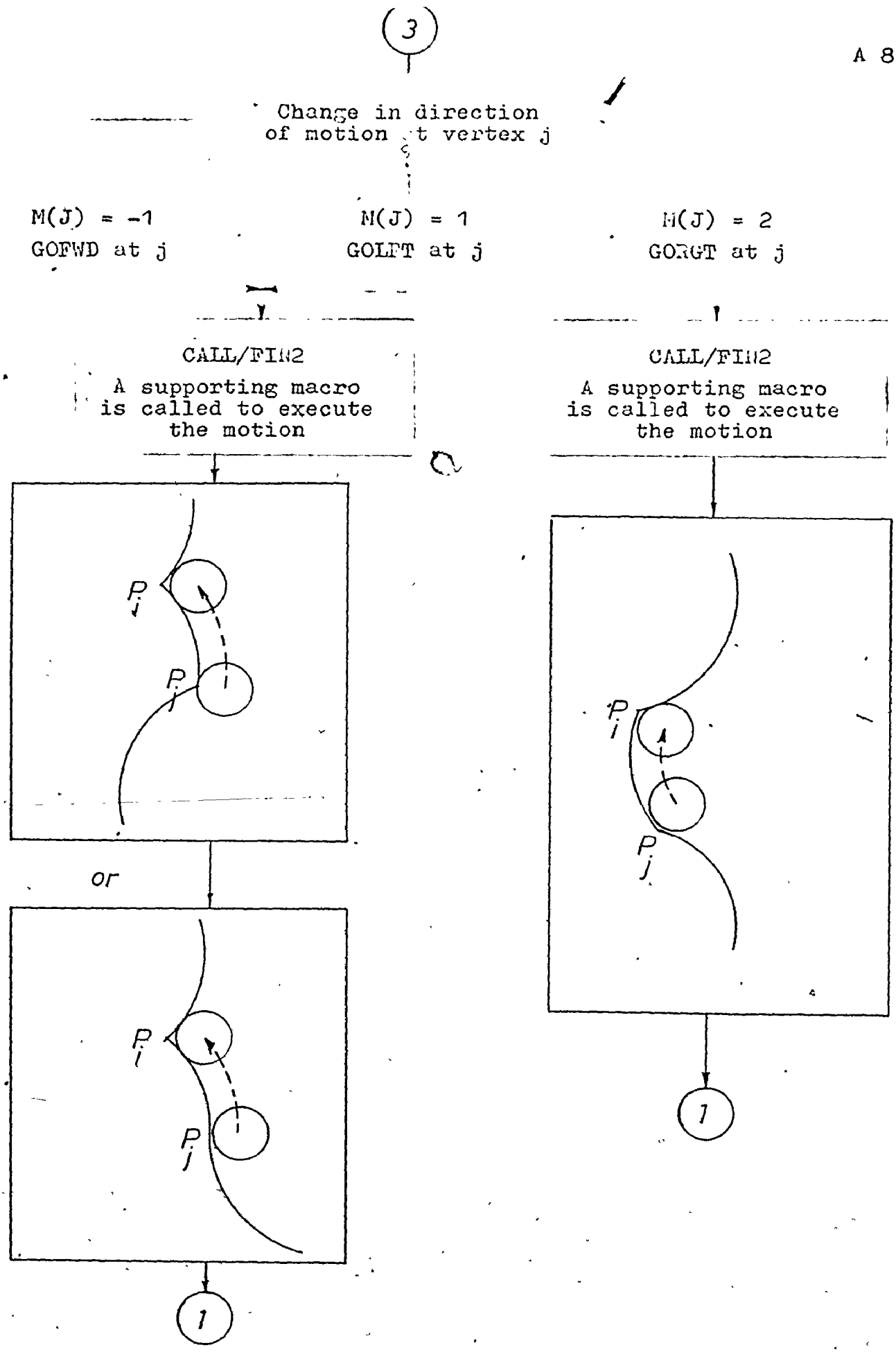
or

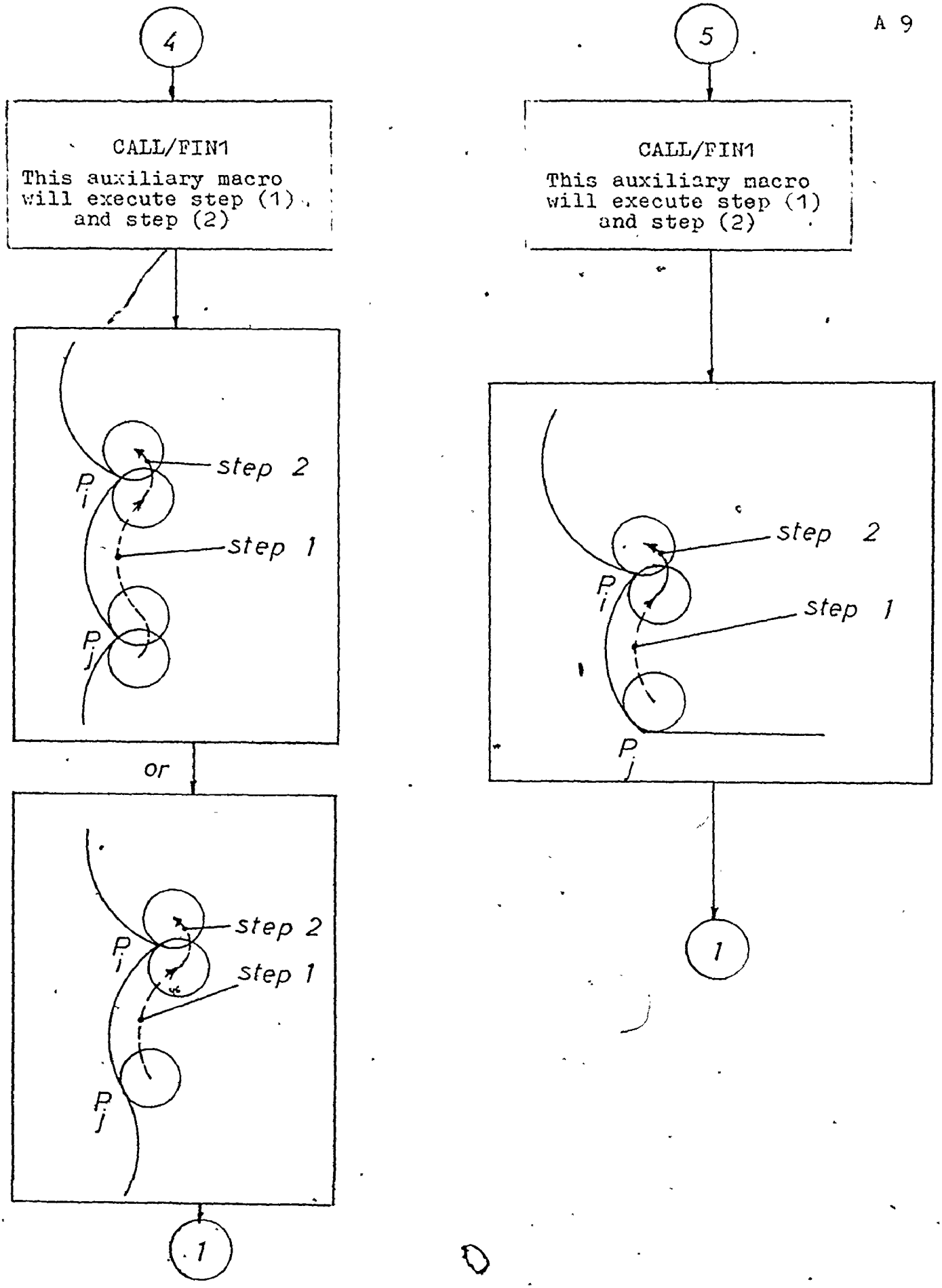


1



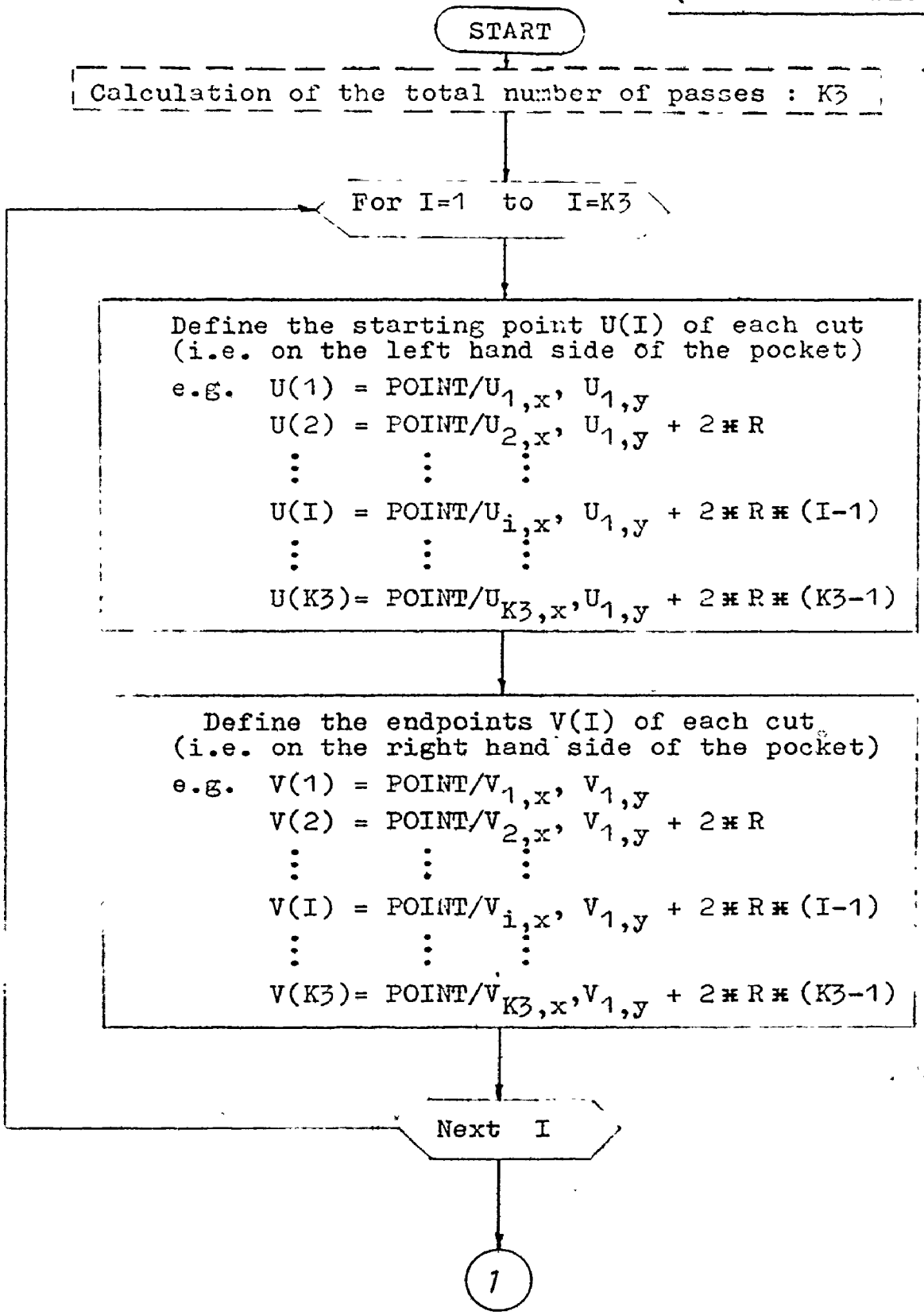
1

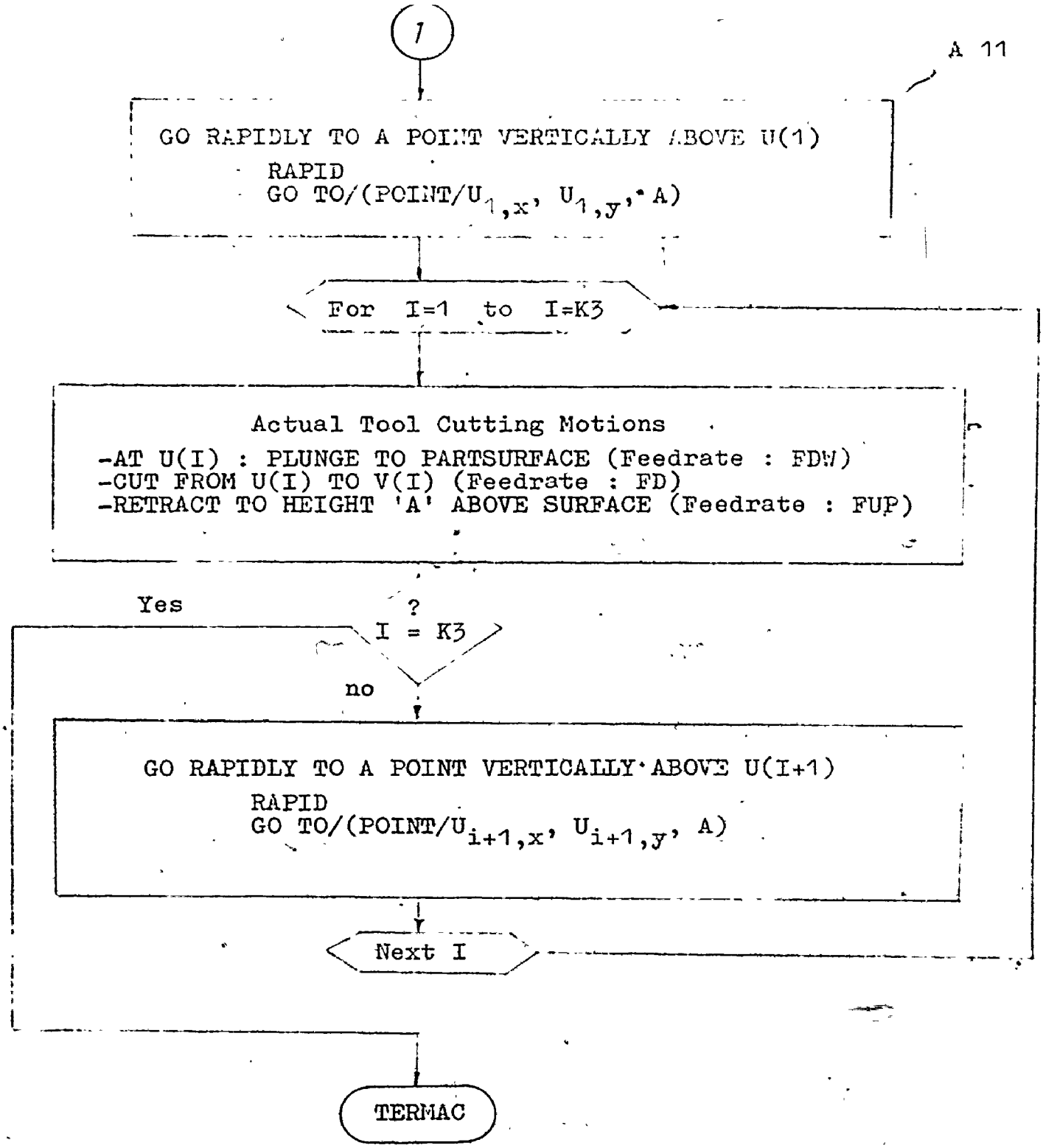

3
Change in direction of motion at vertex j
M(J) = -1
GOFWD at j
M(J) = 1
GOLFT at j
M(J) = 2
GORGT at j
CALL/FIN2
A supporting macro is called to execute the motion
CALL/FIN2
A supporting macro is called to execute the motion
Pi
Pj
or
Pi
Pj
Pi
Pj
1
1


4
CALL/FIN1
This auxiliary macro will execute step (1) and step (2)
step 2
step 1
Pi
Pj
or
step 2
step 1
Pi
Pj
1
5
CALL/FIN1
This auxiliary macro will execute step (1) and step (2)
step 2
step 1
Pi
Pj
1

1.1.3.2. Flowchart RGH1 (unidirectional roughing macro)

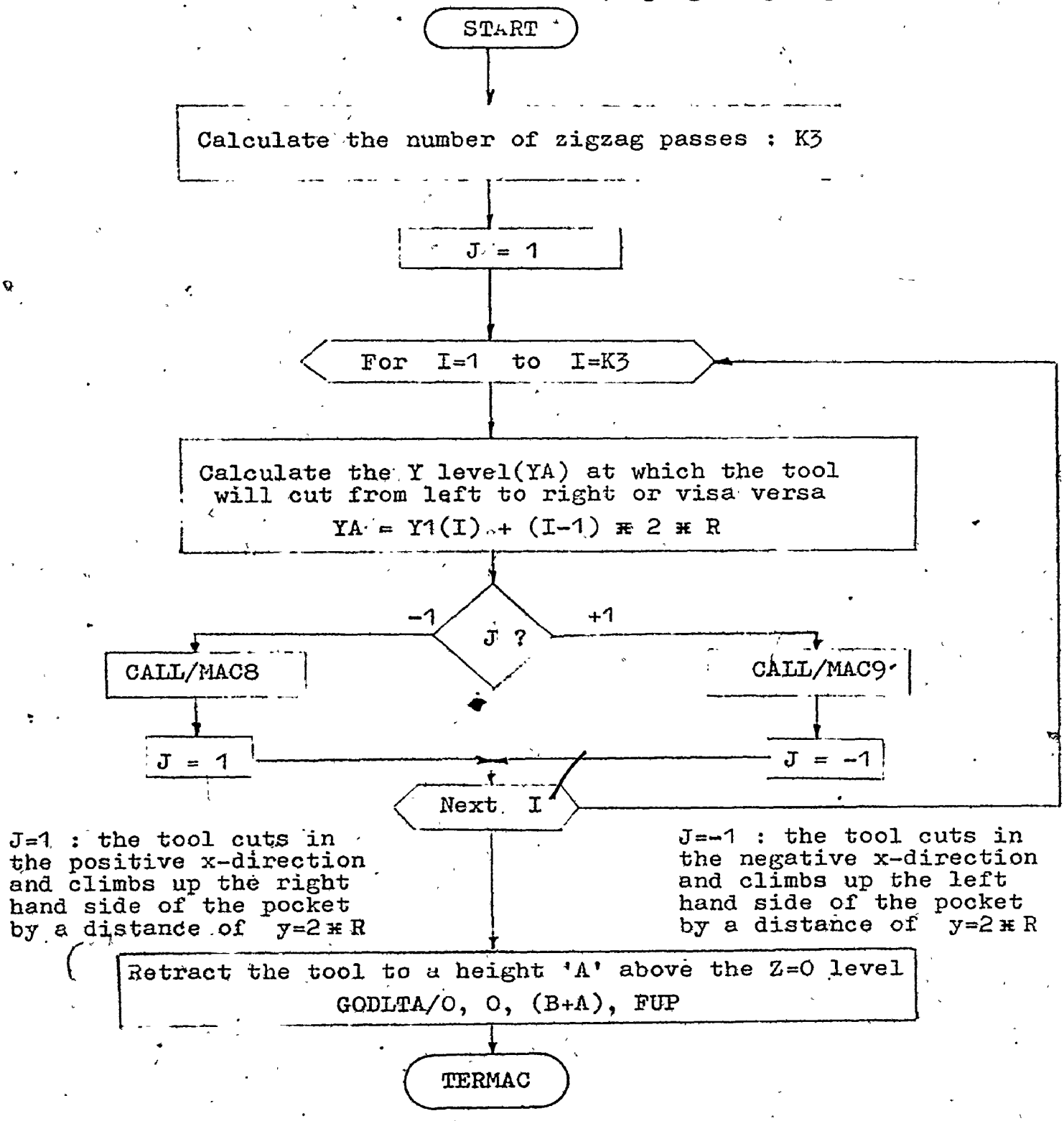
(note : see FIG 3.3.17)



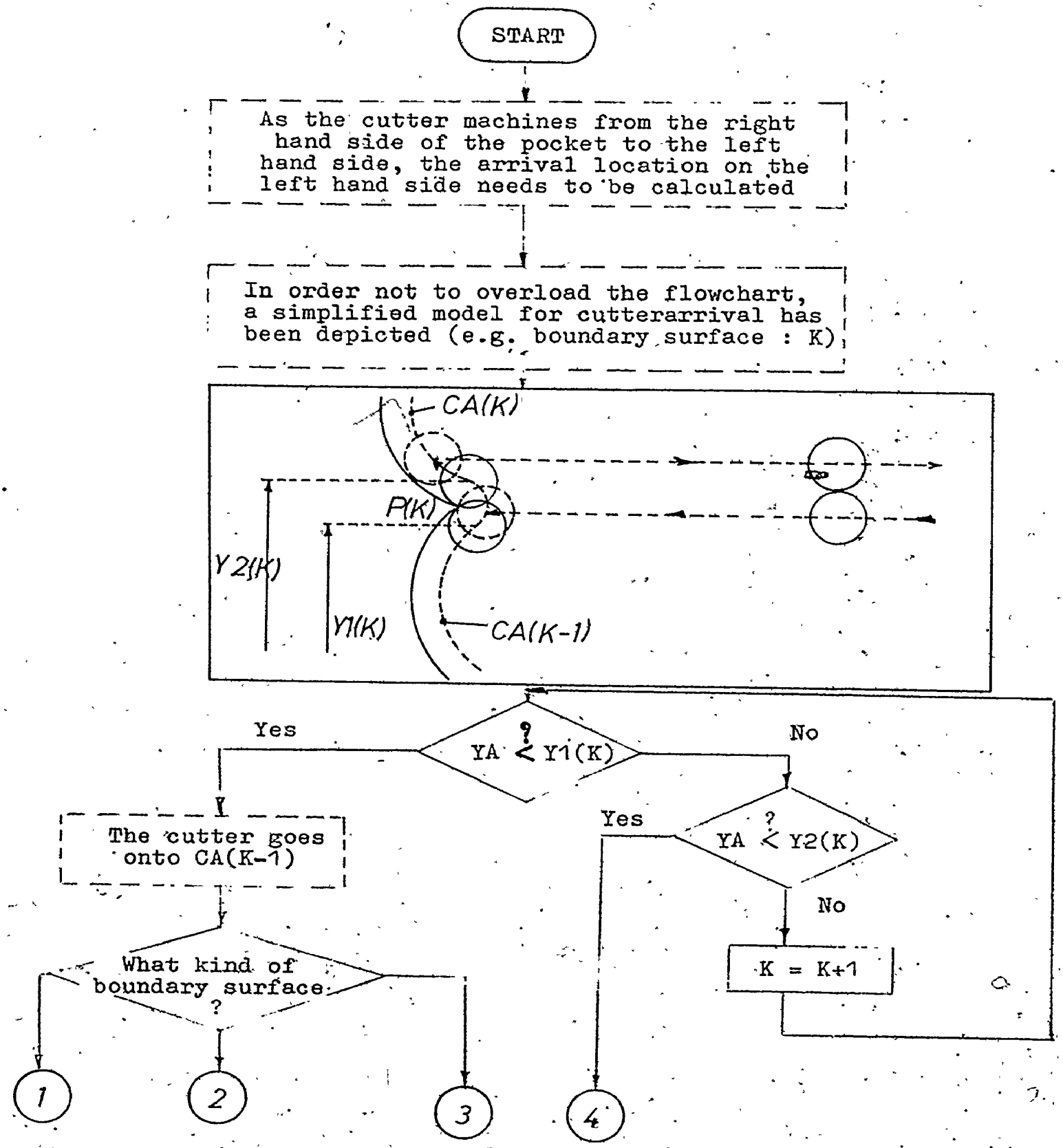

1
GO RAPIDLY TO A POINT VERTICALLY ABOVE U(1)
RAPID
GO TO/(POINT/U1,x, U1,y, A)
For I=1 to I=K3
Actual Tool Cutting Motions
-AT U(I) : PLUNGE TO PARTSURFACE (Feedrate : FDW)
-CUT FROM U(I) TO V(I) (Feedrate : FD)
-RETRACT TO HEIGHT 'A' ABOVE SURFACE (Feedrate : FUP)
Yes
I = K3 ?
no
GO RAPIDLY TO A POINT VERTICALLY ABOVE U(I+1)
RAPID
GO TO/(POINT/Ui+1,x, Ui+1,y, A)
Next I
TERMAC

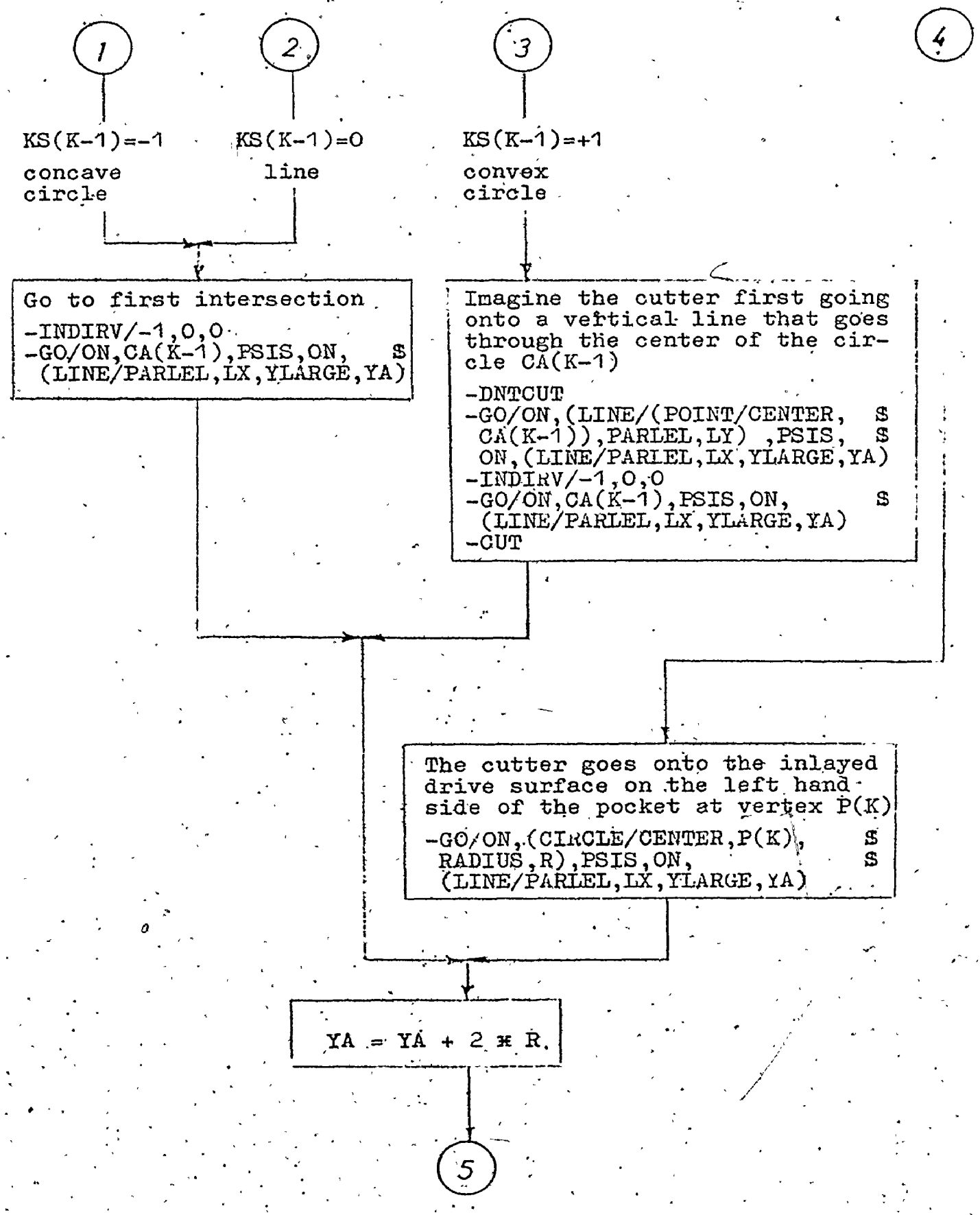
1.1.3.3. Flowchart RGH2 and auxiliary macros MAC8,MAC9,MAC82,MAC83,MAC84,MAC92,MAC93,MAC94,ADD1

(zigzag roughing macro)



Flowchart MAC8
and auxiliary macros MAC82, MAC83, MAC84

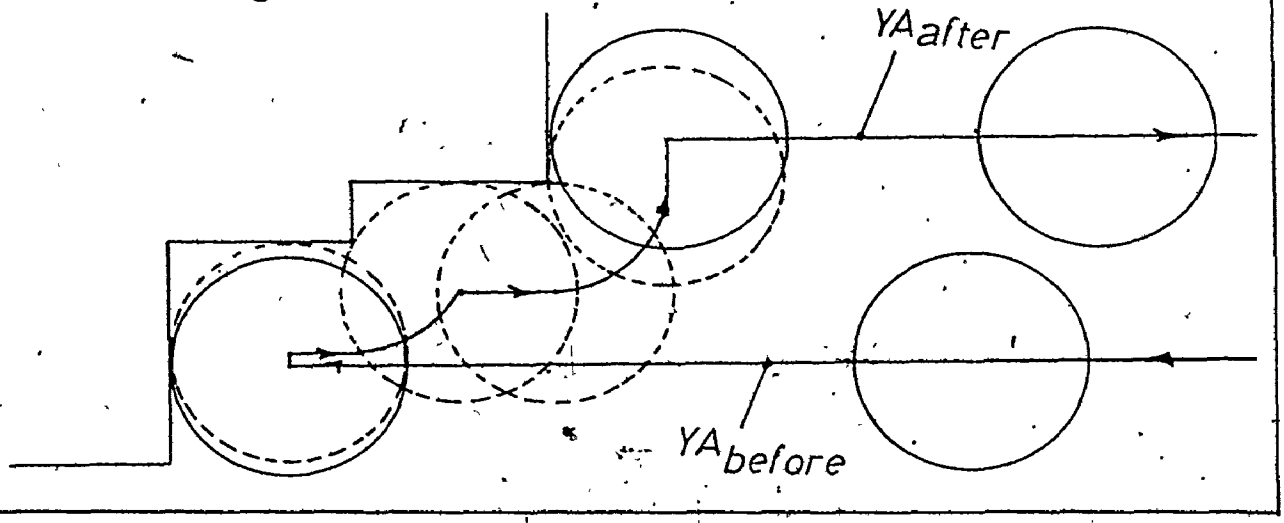



1
2
3
4
KS(K-1)=-1
concave
circle
KS(K-1)=0
line
KS(K-1)=+1
convex
circle
Go to first intersection
-INDIRV/-1,0,0
-GO/ON,CA(K-1),PSIS,ON, $
(LINE/PARLEL,LX,YLARGE,YA)
Imagine the cutter first going onto a vertical line that goes through the center of the circle CA(K-1)
-DNTCUT
-GO/ON,(LINE/(POINT/CENTER, $
CA(K-1)),PARLEL,LY) ,PSIS, $
ON,(LINE/PARLEL,LX,YLARGE,YA)
-INDIRV/-1,0,0
-GO/ON,CA(K-1),PSIS,ON, $
(LINE/PARLEL,LX,YLARGE,YA)
-CUT
The cutter goes onto the inlayed drive surface on the left hand side of the pocket at vertex P(K)
-GO/ON,(CIRCLE/CENTER,P(K), $
RADIUS,R),PSIS,ON, $
(LINE/PARLEL,LX,YLARGE,YA)
YA = YA + 2 * R.
5

5

Climb up the left hand side of the pocket so that Y-cutter becomes YA

e.g.

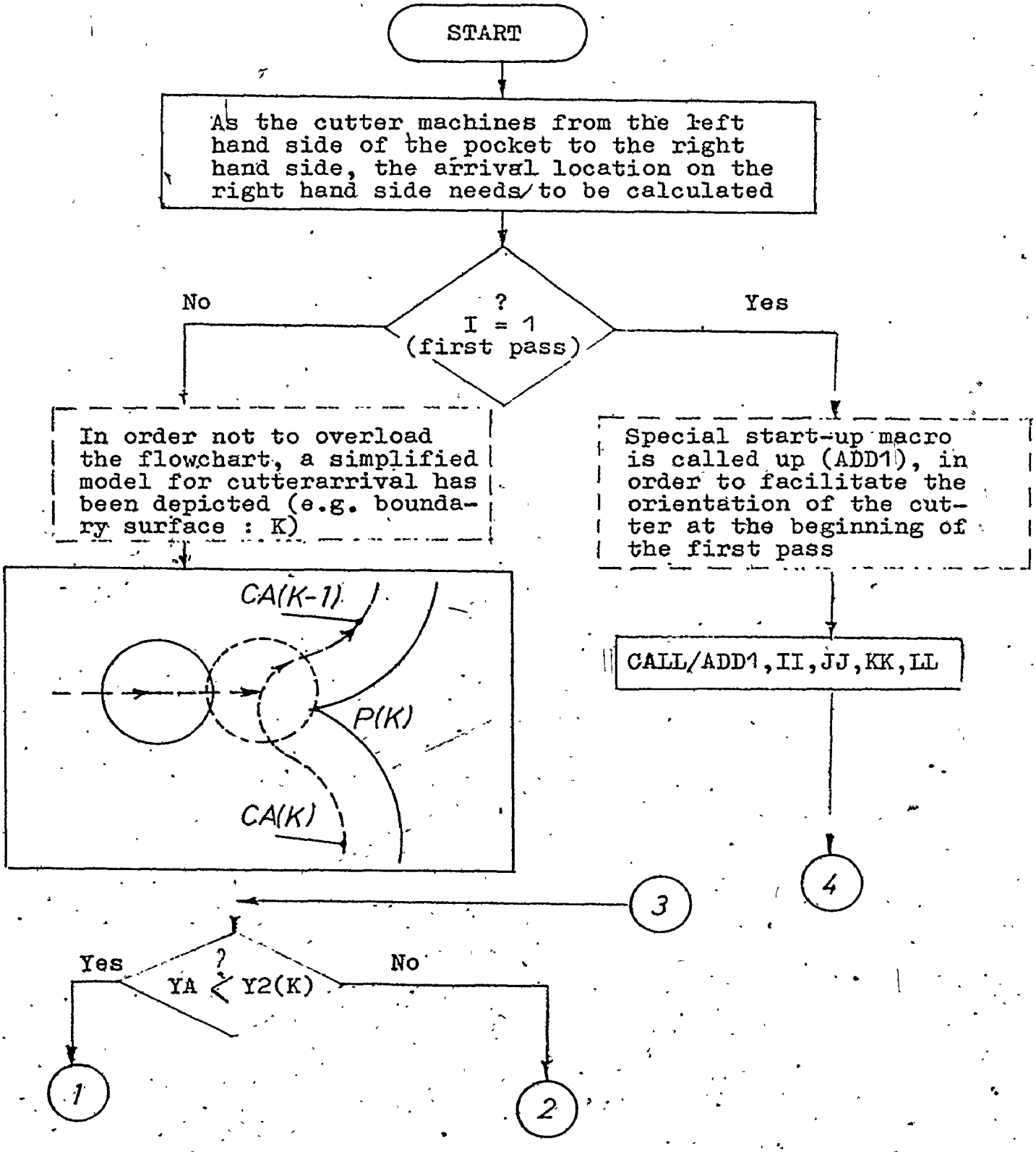


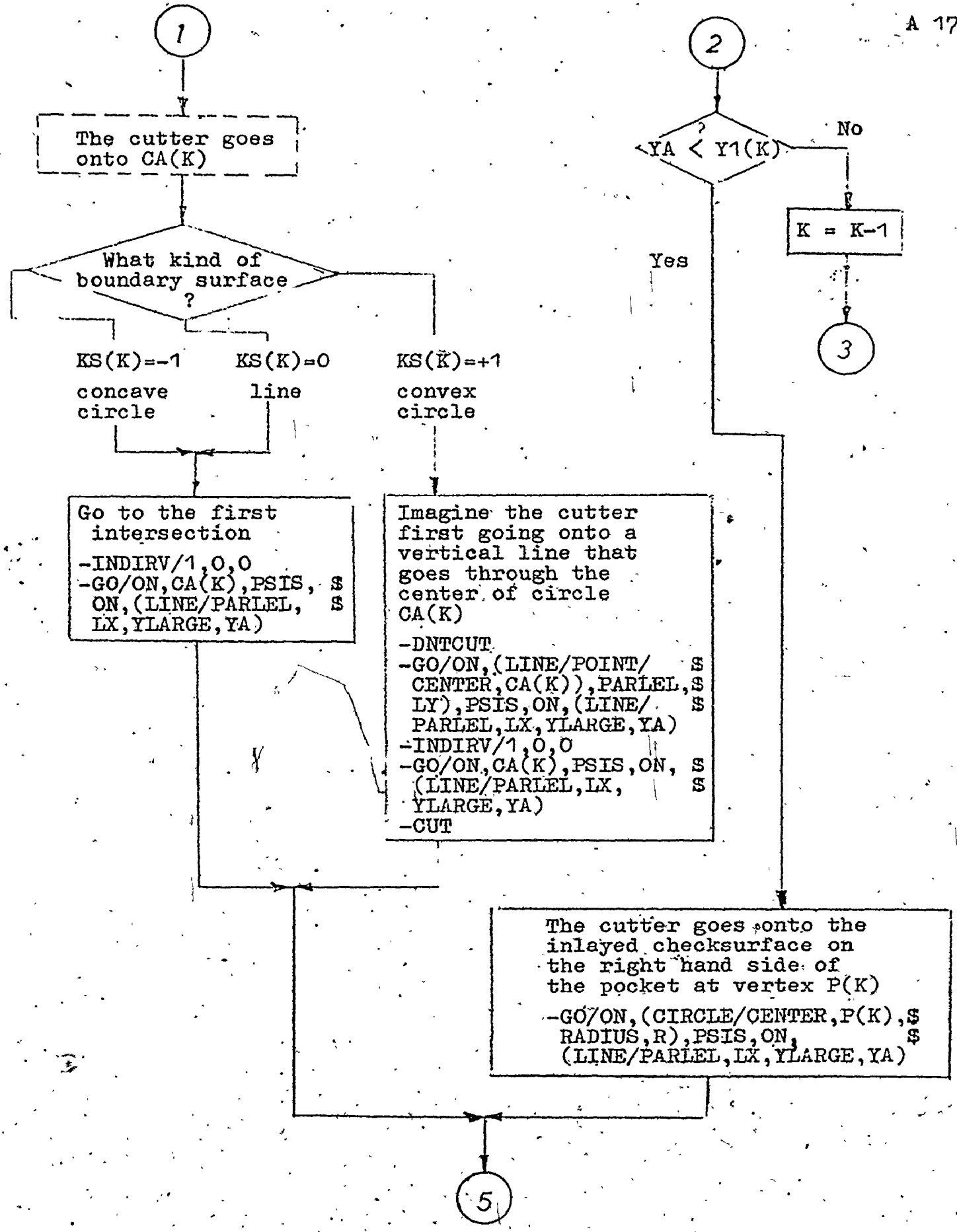
In order to accomplish this supporting macros need to be called up

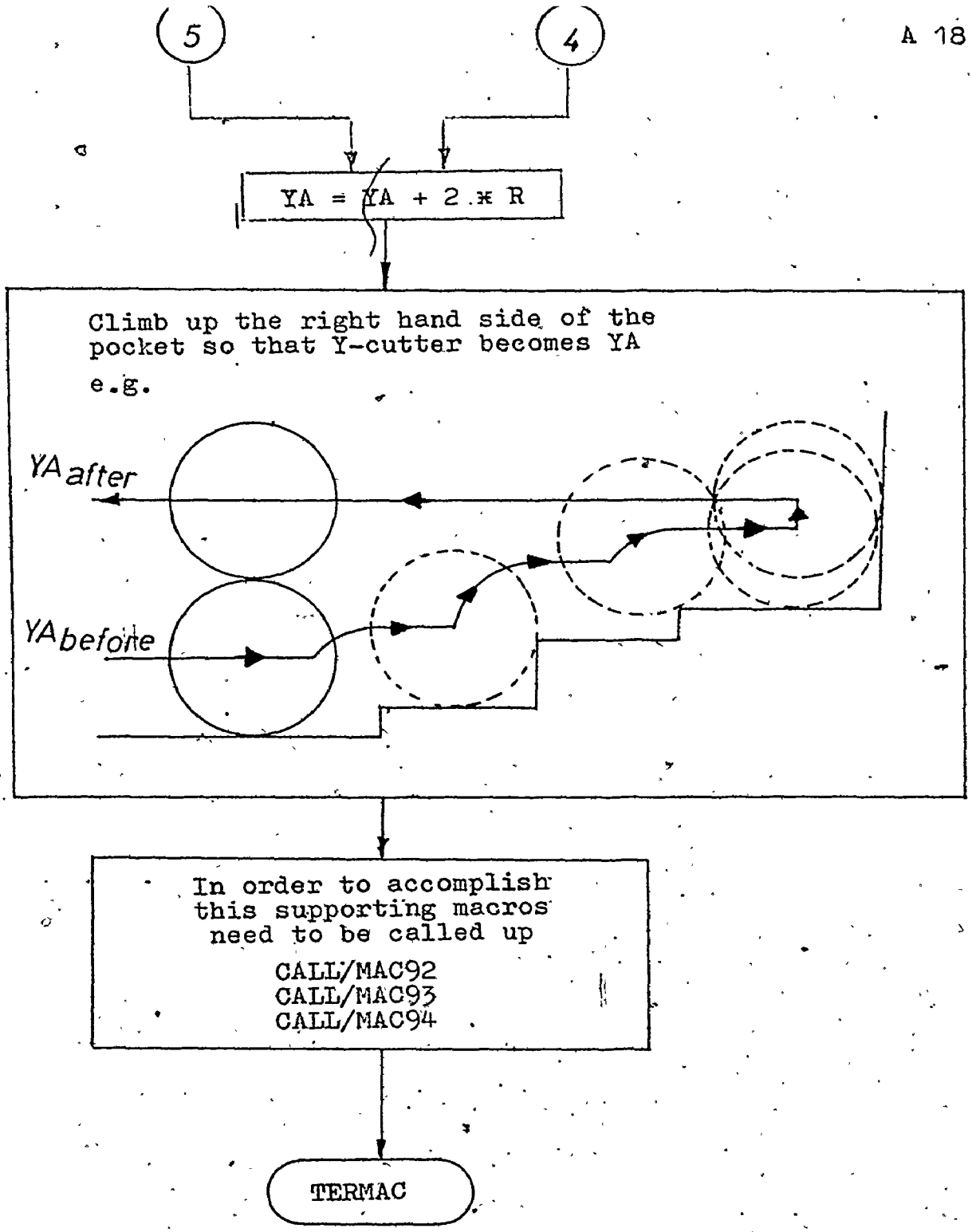
CALL/MAC82
CALL/MAC83
CALL/MAC84

TERMAC

Flowchart MAC9
and auxiliary macros MAC92,MAC93,MAC94,ADD1




1
The cutter goes onto CA(K)
What kind of boundary surface ?
KS(K)=-1
concave circle
KS(K)=0
line
KS(K)=+1
convex circle
Go to the first intersection
-INDIRV/1,0,0
-GO/ON,CA(K),PSIS, $
ON,(LINE/PARLEL, $
LX,YLARGE,YA)
Imagine the cutter first going onto a vertical line that goes through the center of circle CA(K)
-DNTCUT
-GO/ON,(LINE/POINT/ $
CENTER,CA(K)),PARLEL,$
LY),PSIS,ON,(LINE/ $
PARLEL,LX,YLARGE,YA)
-INDIRV/1,0,0
-GO/ON,CA(K),PSIS,ON, $
(LINE/PARLEL,LX, $
YLARGE,YA)
-CUT
2
YA < Y1(K) ?
No
K = K-1
3
Yes
The cutter goes onto the inlayed checksurface on the right hand side of the pocket at vertex P(K)
-GO/ON,(CIRCLE/CENTER,P(K),$
RADIUS,R),PSIS,ON, $
(LINE/PARLEL,LX,YLARGE,YA)
5


5
4
YA = YA + 2.* R
Climb up the right hand side of the pocket so that Y-cutter becomes YA
e.g.
YA after
YA before
In order to accomplish this supporting macros need to be called up
CALL/MAC92
CALL/MAC93
CALL/MAC94
TERMAC

1.1.3.4. Flowchart SPIRAL

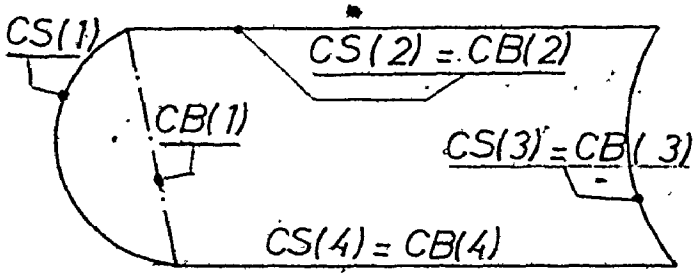
START

CALL/SPIR2, J=1
CALL/SPIR2, J=3

CALL/SPIR4 → 1

Macro SPIR2

- determines the auxiliary boundary surfaces CB(J) when the first or third surface is convex



- CALL/SPIR1, JJ=J
- CALL/SPIR3, JJ=J, AA=XLARGE

Macro SPIR4

-the smallest inner angle is determined AX

-the width of cut is calculated in order not to have any cusps left-over when machining with cutterradius R

width = $2 * R * \sin(\frac{AX}{2})$

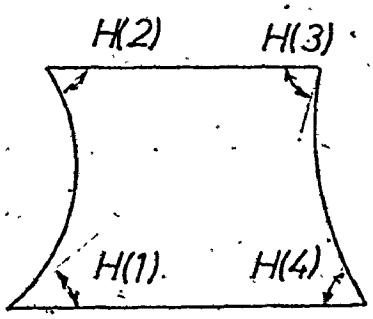
e.g. for a rectangular pocket : AX=90°
width = .854 R

-the number of quasi-circles in the spiral are being calculated

-e.g. CALL/SPIR5, $
JJ=1, AA=XLARGE .

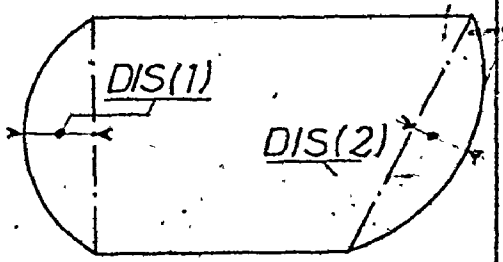
Macro SPIR1

The inner angles H(J) and H(J+1) are determined



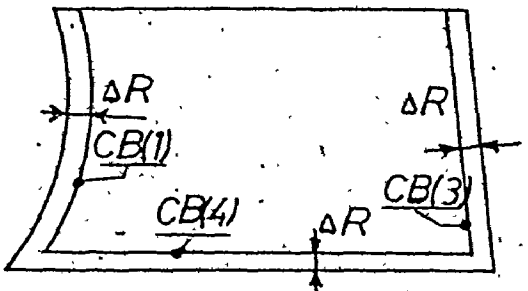
Macro SPIR3

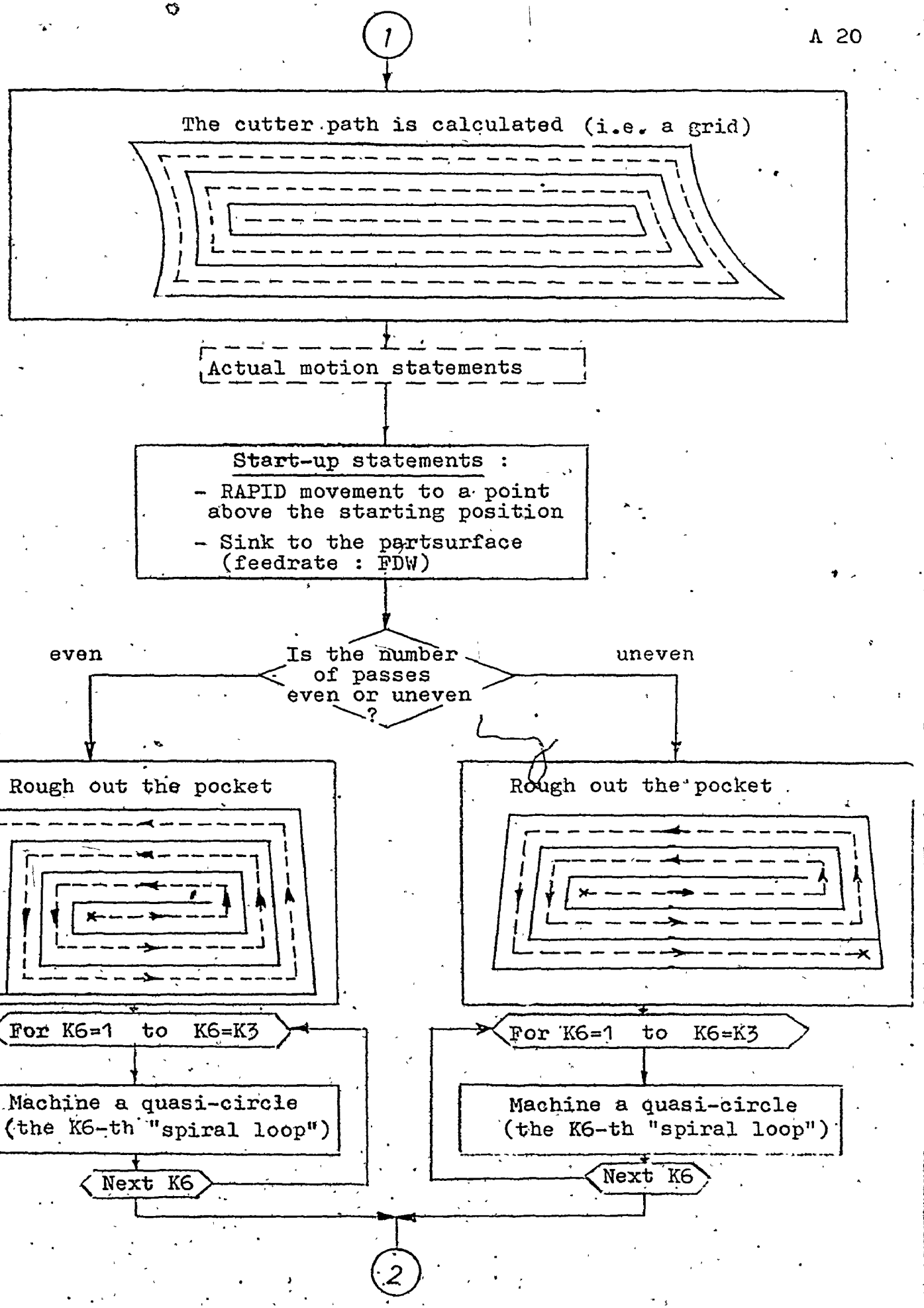
The maximum distance in the left-over areas is being calculated

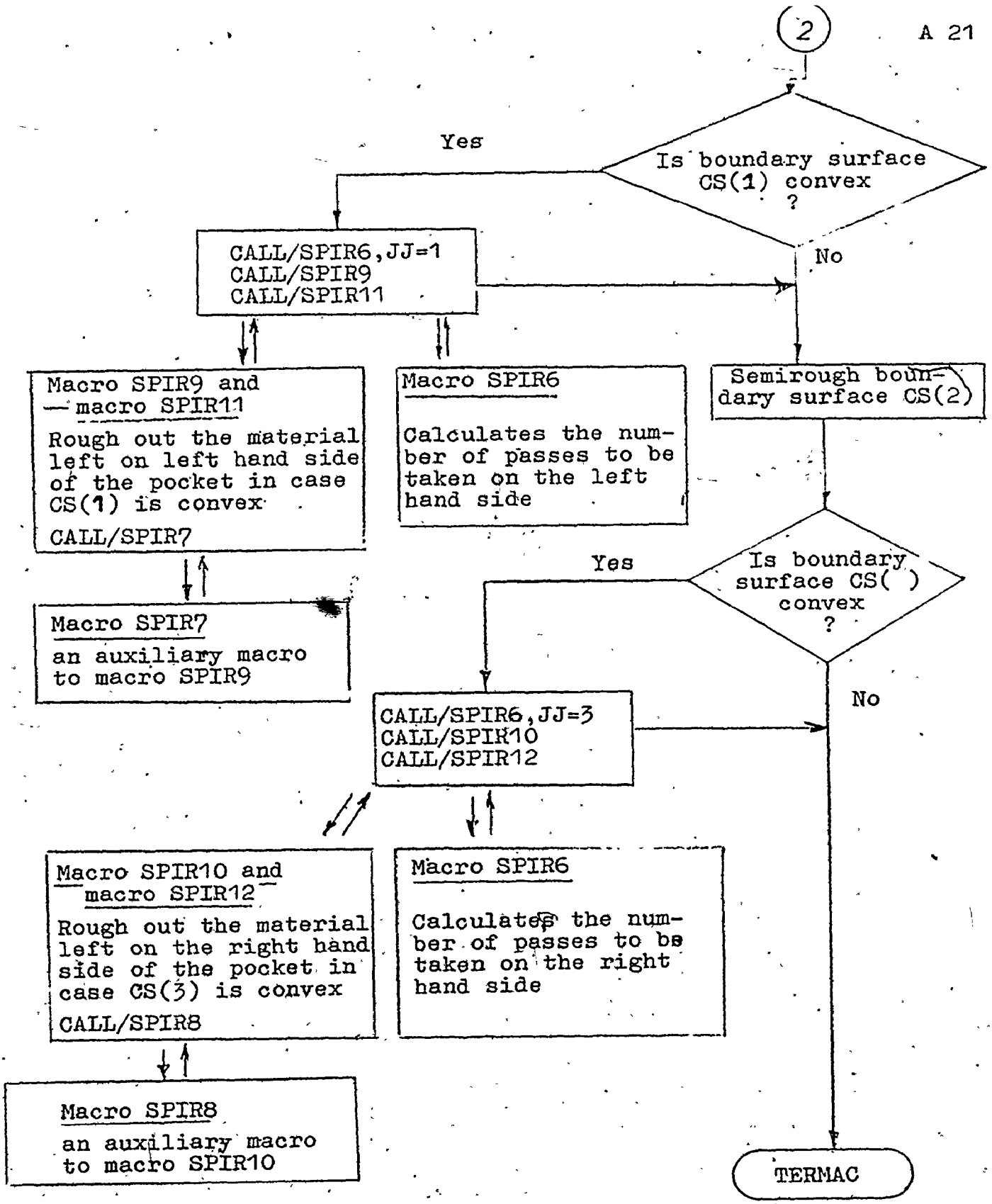


Macro SPIR5

New offset boundary surfaces are calculated CB(J)



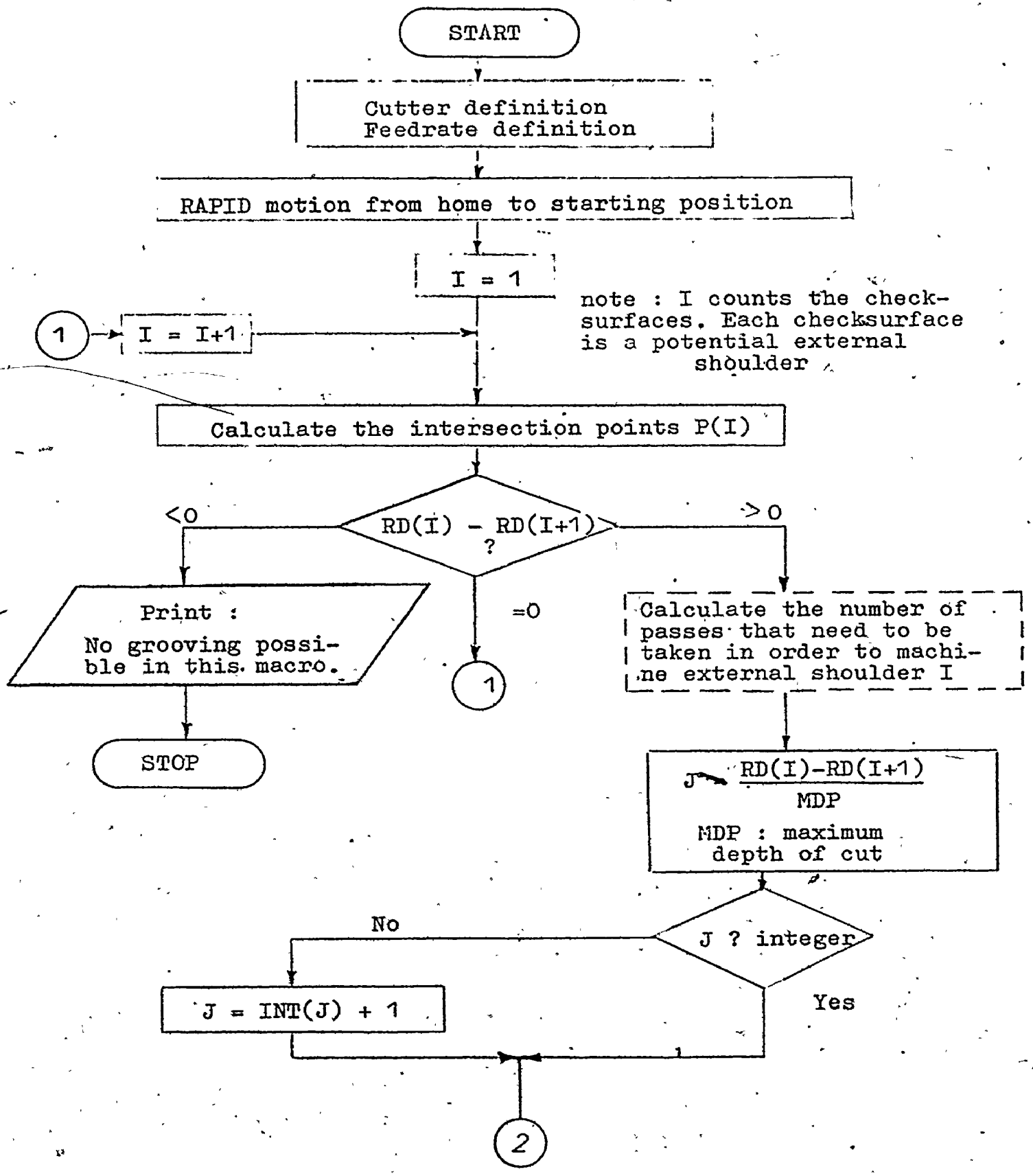

1
The cutter path is calculated (i.e. a grid)
Actual motion statements
Start-up statements :
- RAPID movement to a point above the starting position
- Sink to the partsurface (feedrate : FDW)
Is the number of passes even or uneven ?
even
uneven
Rough out the pocket
Rough out the pocket
For K6=1 to K6=K3
For K6=1 to K6=K3
Machine a quasi-circle (the K6-th "spiral loop")
Machine a quasi-circle (the K6-th "spiral loop")
Next K6
Next K6
2

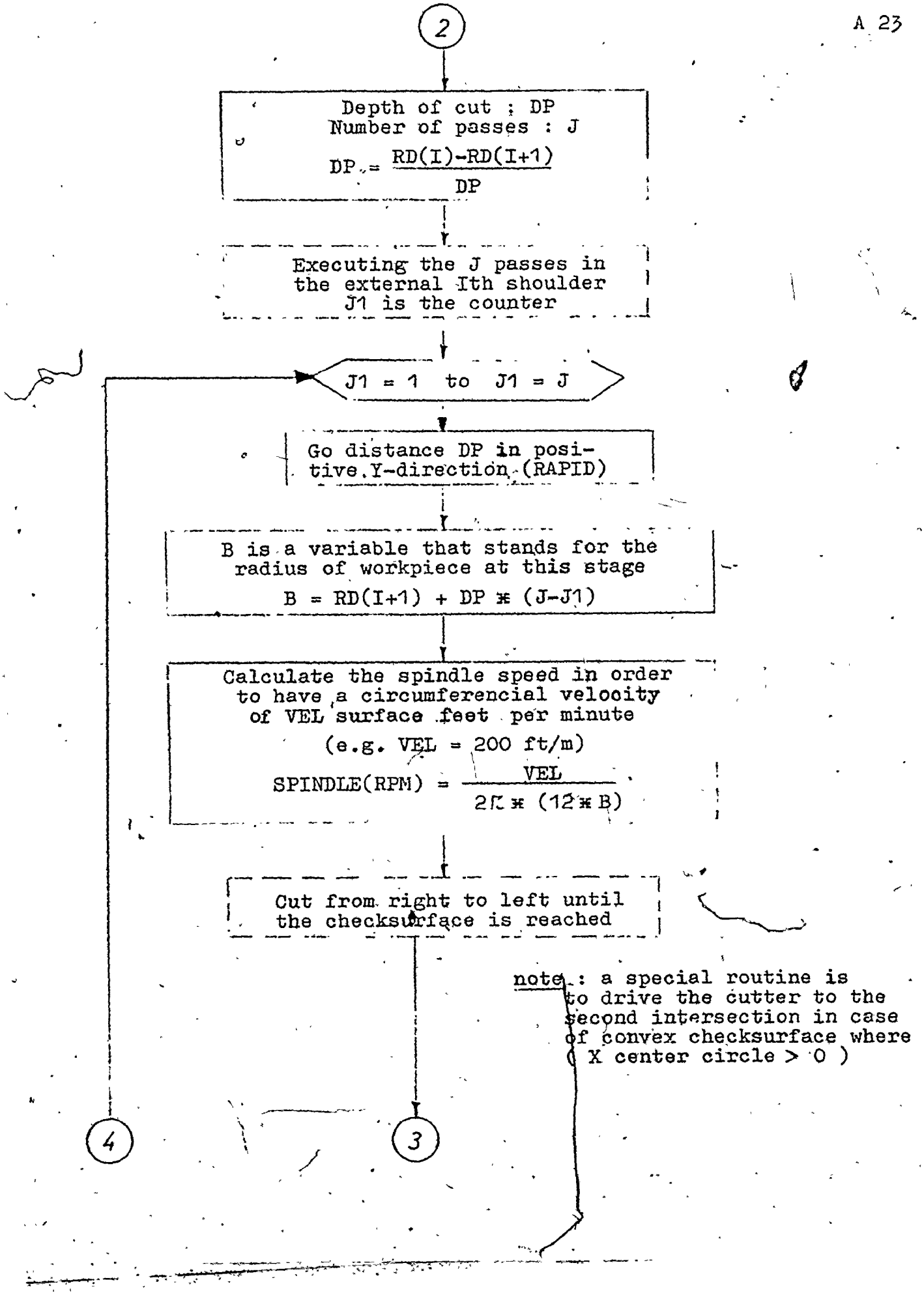

2
Is boundary surface CS(1) convex ?
Yes
No
CALL/SPIR6,JJ=1
CALL/SPIR9
CALL/SPIR11
Macro SPIR9 and macro SPIR11
Rough out the material left on left hand side of the pocket in case CS(1) is convex
CALL/SPIR7
Macro SPIR6
Calculates the number of passes to be taken on the left hand side
Semirough boundary surface CS(2)
Macro SPIR7
an auxiliary macro to macro SPIR9
Is boundary surface CS( ) convex ?
Yes
No
CALL/SPIR6,JJ=3
CALL/SPIR10
CALL/SPIR12
Macro SPIR10 and macro SPIR12
Rough out the material left on the right hand side of the pocket in case CS(3) is convex
CALL/SPIR8
Macro SPIR6
Calculates the number of passes to be taken on the right hand side
Macro SPIR8
an auxiliary macro to macro SPIR10
TERMAC

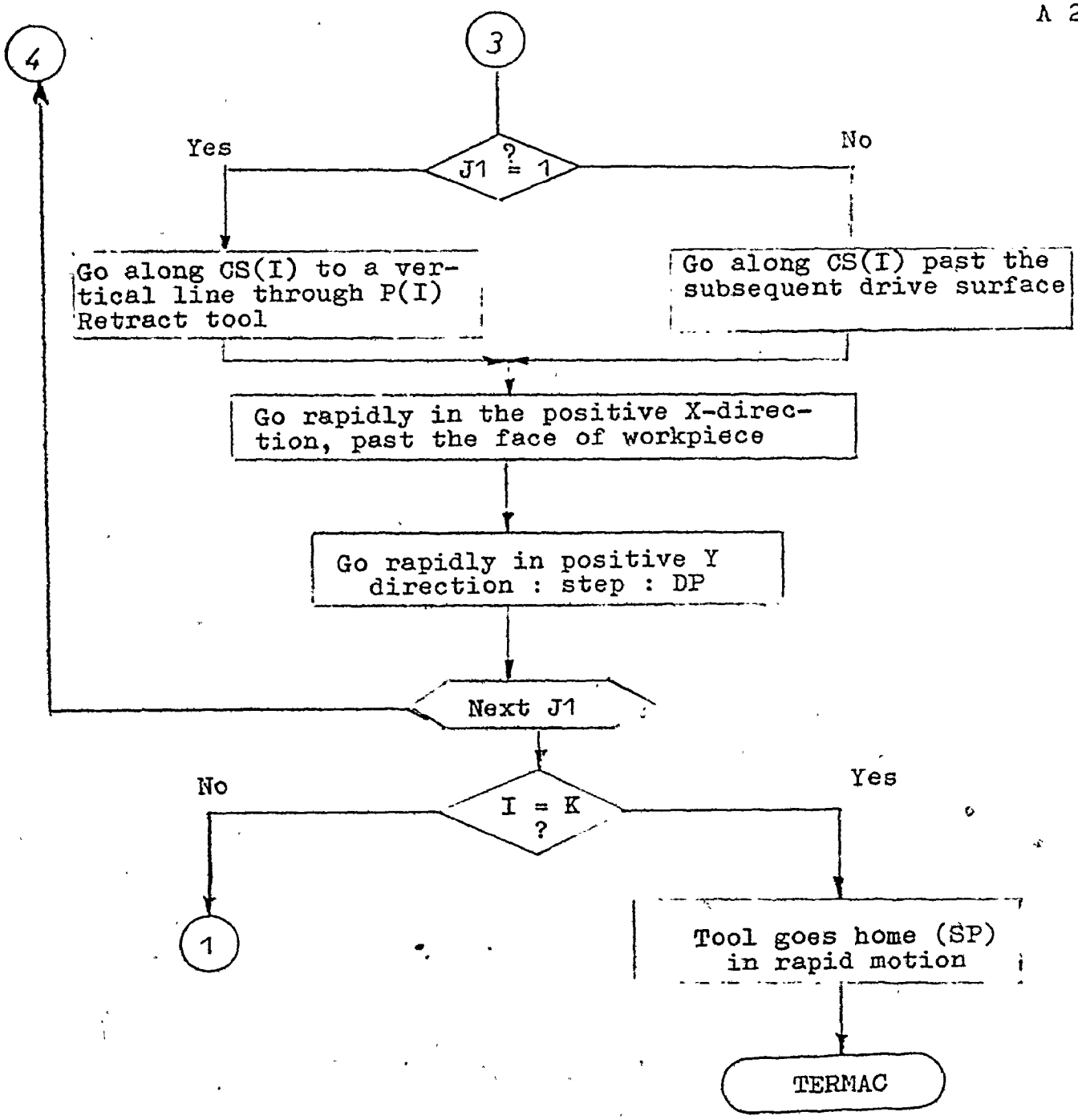
1.2. The turning macros

1.2.1. Flowchart ROUGH (Barstock turning macro)

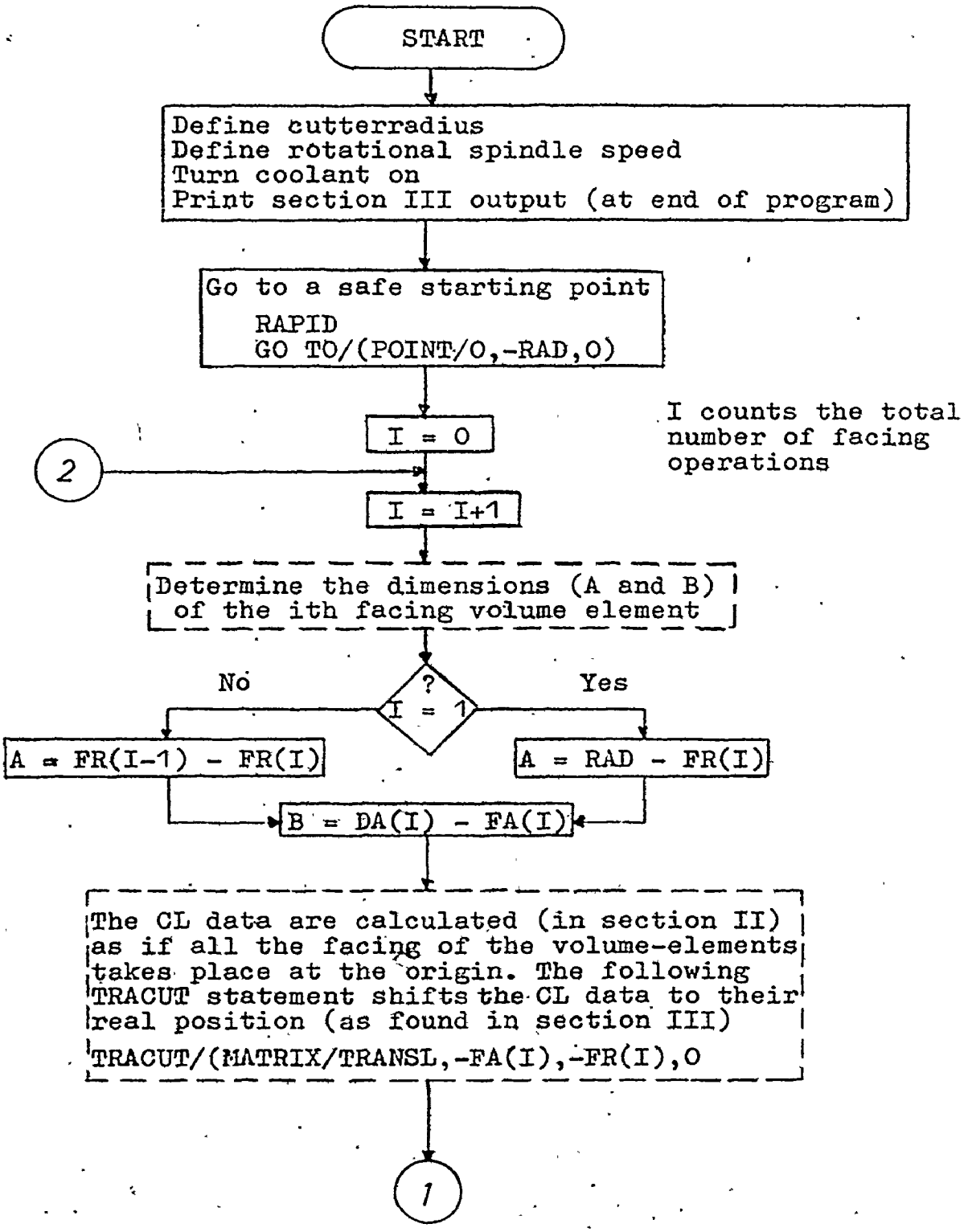
(note : see FIG 3.3.25)

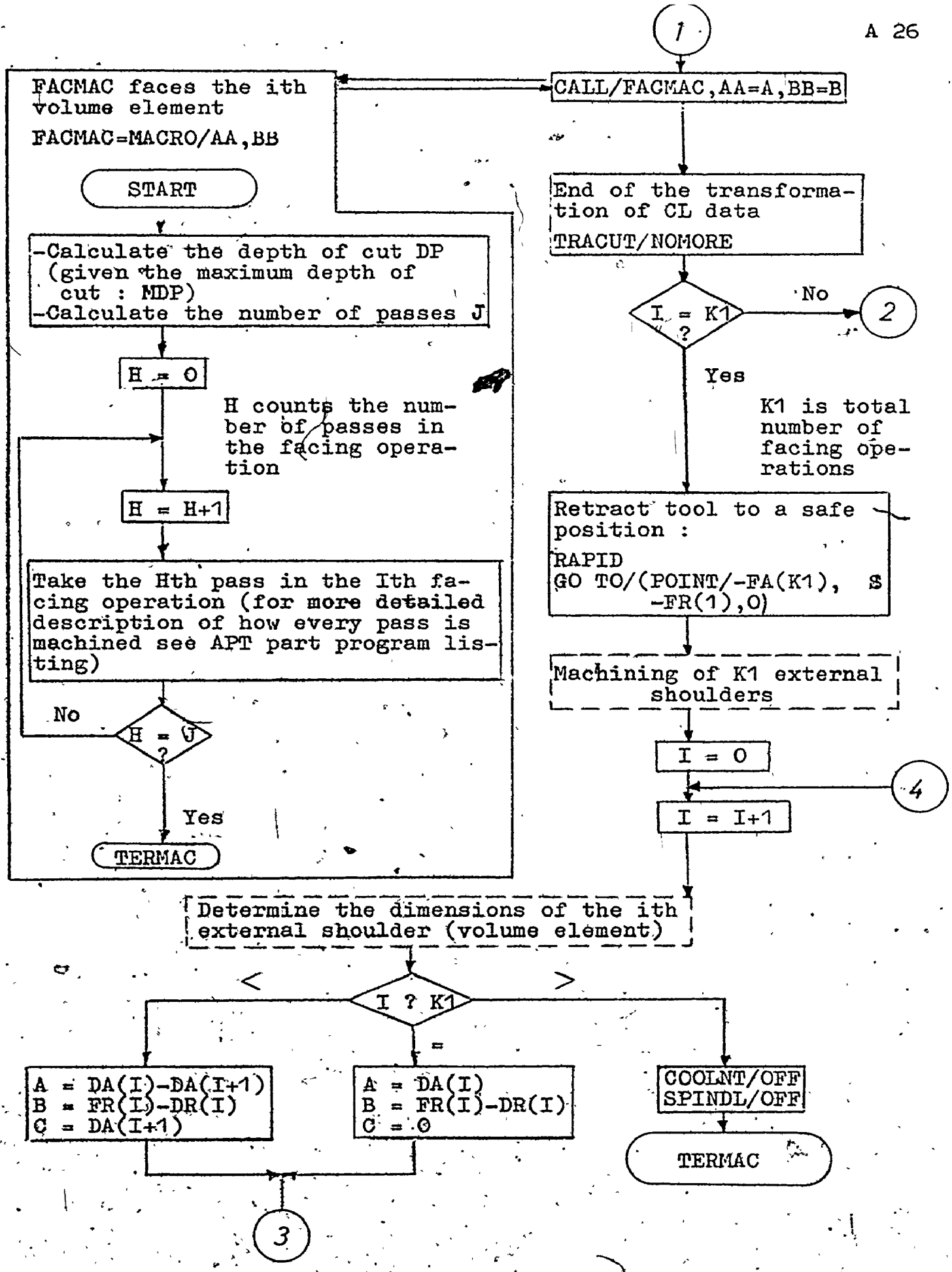


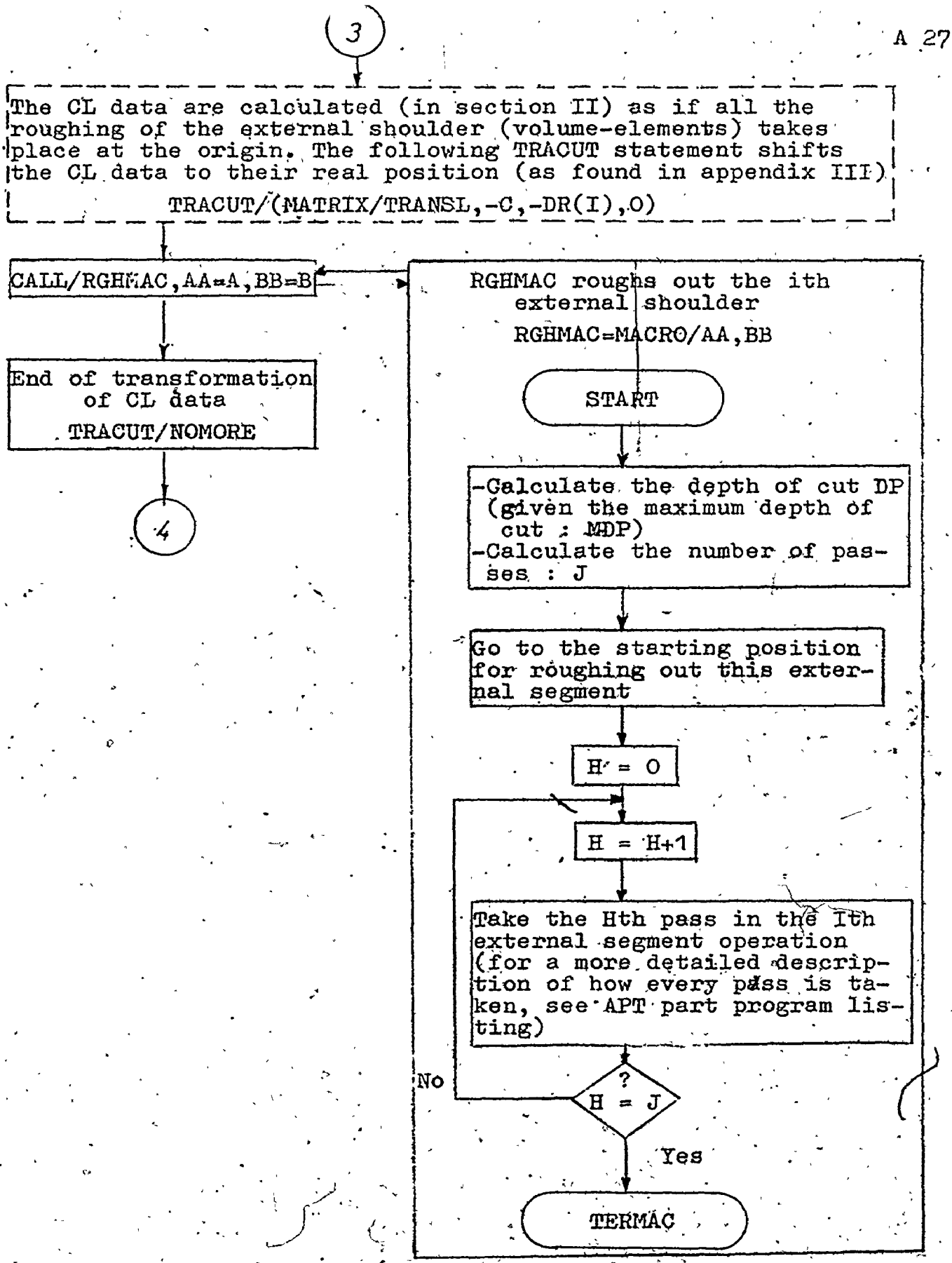

2
Depth of cut : DP
Number of passes : J
DP = RD(I)-RD(I+1) / DP
Executing the J passes in the external Ith shoulder J1 is the counter
J1 = 1 to J1 = J
Go distance DP in positive Y-direction (RAPID)
B is a variable that stands for the radius of workpiece at this stage
B = RD(I+1) + DP * (J-J1)
Calculate the spindle speed in order to have a circumferencial velocity of VEL surface feet per minute
(e.g. VEL = 200 ft/m)
SPINDLE(RPM) = VEL / 2π * (12 * B)
Cut from right to left until the checksurface is reached
note : a special routine is to drive the cutter to the second intersection in case of convex checksurface where (X center circle > 0)
4
3


4
3
Yes
J1 ? = 1
No
Go along CS(I) to a vertical line through P(I) Retract tool
Go along CS(I) past the subsequent drive surface
Go rapidly in the positive X-direction, past the face of workpiece
Go rapidly in positive Y direction : step : DP
Next J1
No
I = K ?
Yes
1
Tool goes home (SP) in rapid motion
TERMAC

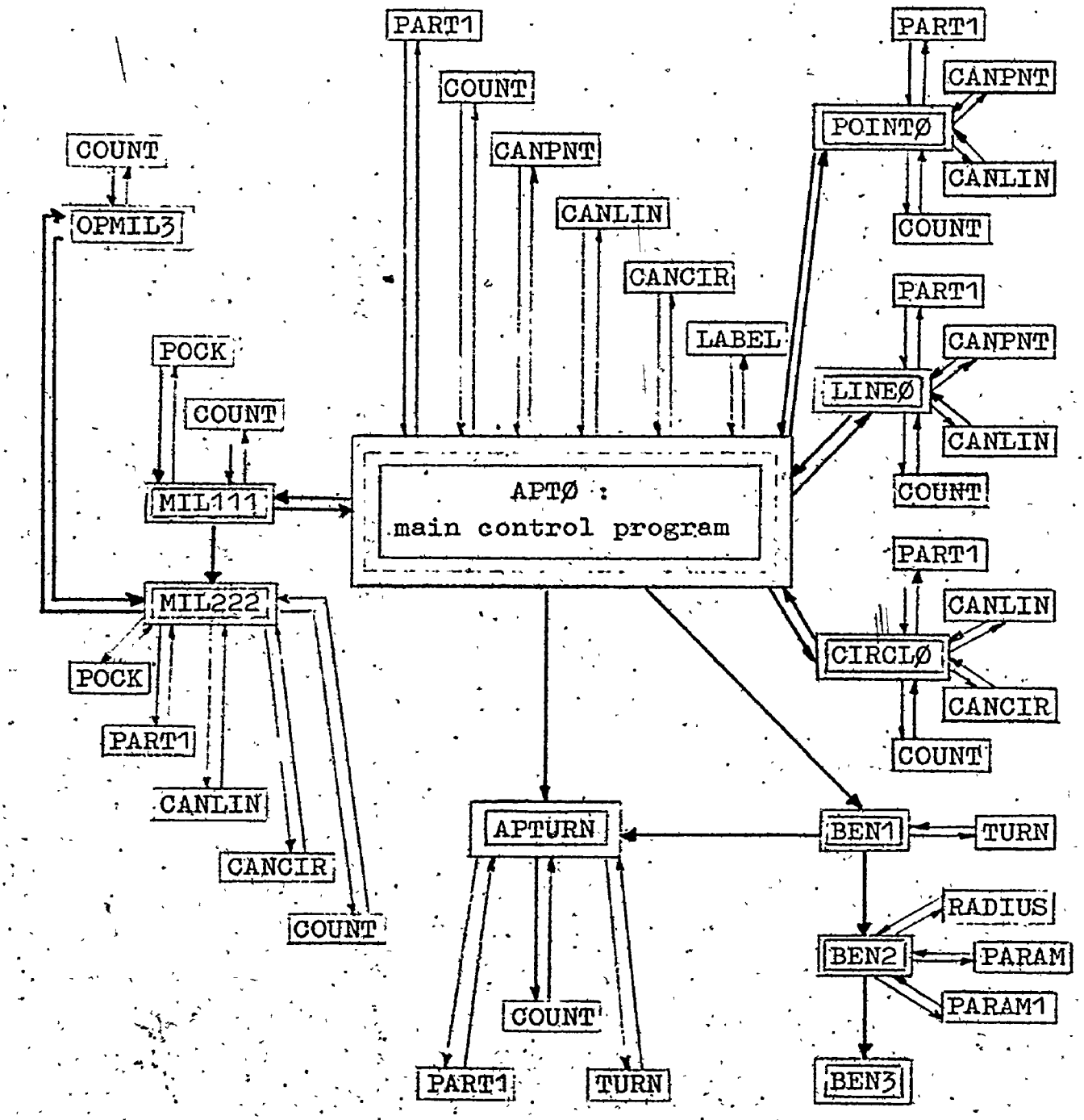
1.2.2. Flowchart SUPV1 and supporting macros FACMAC and RGHMAC (Turning a stepped shaft from a forging)



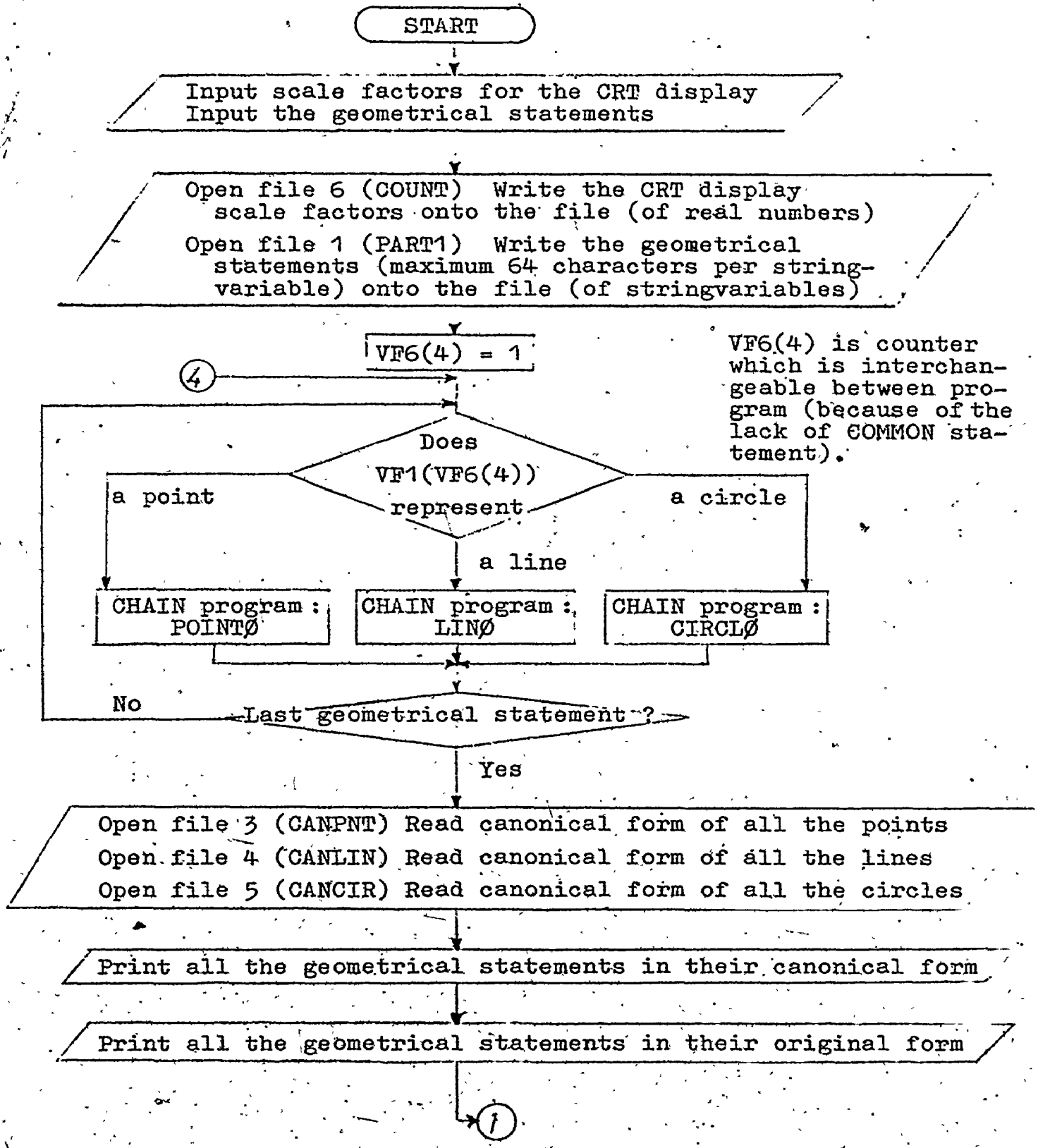

1
CALL/FACMAC,AA=A,BB=B
FACMAC faces the ith volume element
FACMAC=MACRO/AA,BB
START
-Calculate the depth of cut DP (given the maximum depth of cut : MDP)
-Calculate the number of passes J
H = 0
H counts the number of passes in the facing operation
H = H+1
Take the Hth pass in the Ith facing operation (for more detailed description of how every pass is machined see APT part program listing)
No
H = J ?
Yes
TERMAC
End of the transformation of CL data
TRACUT/NOMORE
I = K1 ?
No
2
Yes
K1 is total number of facing operations
Retract tool to a safe position :
RAPID
GO TO/(POINT/-FA(K1), -FR(1),0)
Machining of K1 external shoulders
I = 0
4
I = I+1
Determine the dimensions of the ith external shoulder (volume element)
I ? K1
<
=
>
A = DA(I)-DA(I+1)
B = FR(I)-DR(I)
C = DA(I+1)
A = DA(I)
B = FR(I)-DR(I)
C = 0
COOLNT/OFF
SPINDL/OFF
TERMAC
3

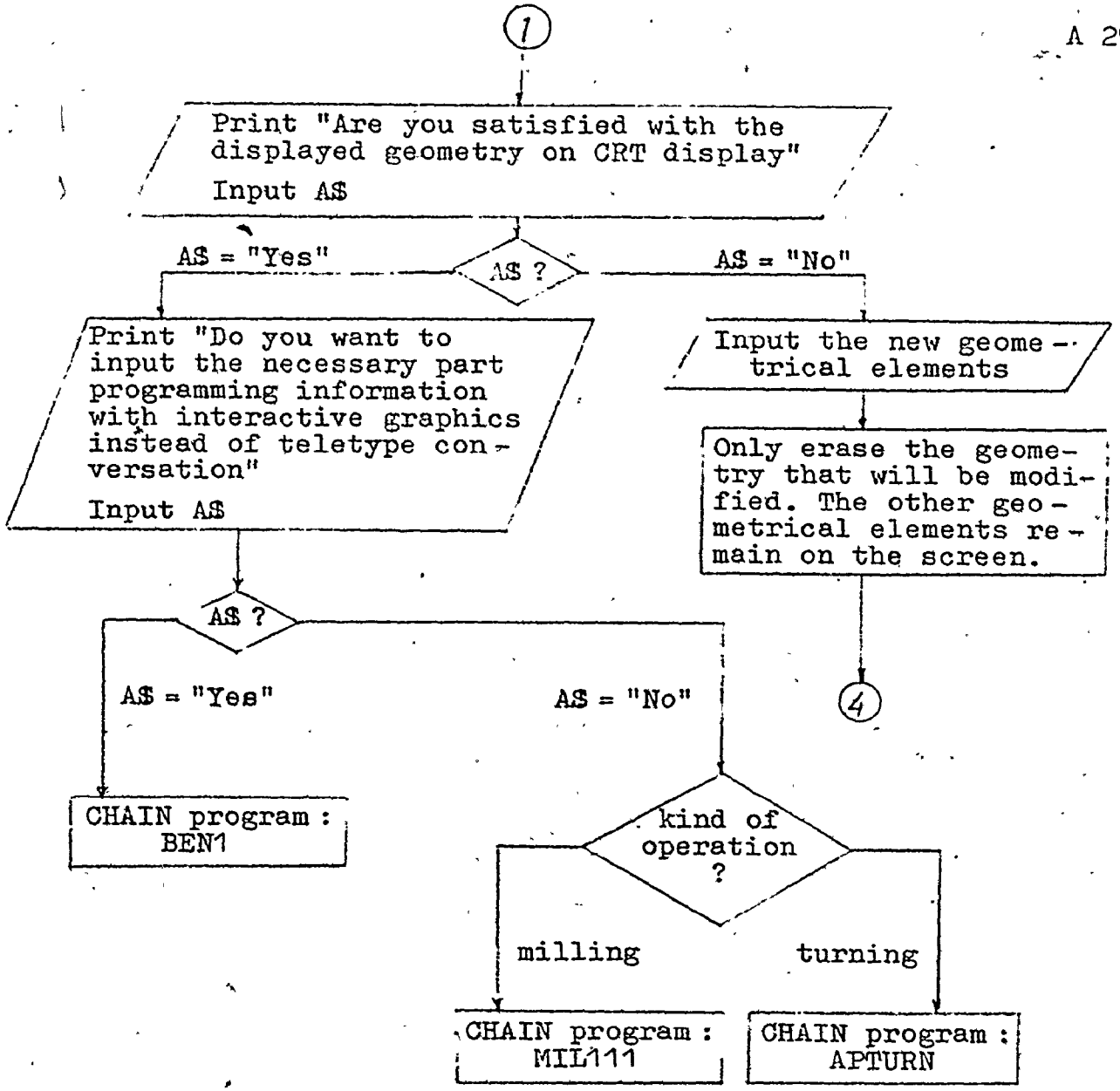

3
The CL data are calculated (in section II) as if all the roughing of the external shoulder (volume-elements) takes place at the origin. The following TRACUT statement shifts the CL data to their real position (as found in appendix III)
TRACUT/(MATRIX/TRANSL,-C,-DR(I),0)
CALL/RGHMAC,AA=A,BB=B
End of transformation of CL data
TRACUT/NOMORE
4
RGHMAC roughs out the ith external shoulder
RGHMAC=MACRO/AA,BB
START
-Calculate the depth of cut DP (given the maximum depth of cut : MDP)
-Calculate the number of passes : J
Go to the starting position for roughing out this external segment
H = 0
H = H+1
Take the Hth pass in the Ith external segment operation (for a more detailed description of how every pass is taken, see APT part program listing)
H = J ?
No
Yes
TERMAC

2. The preprocessor software package architecture.
General lay-out.

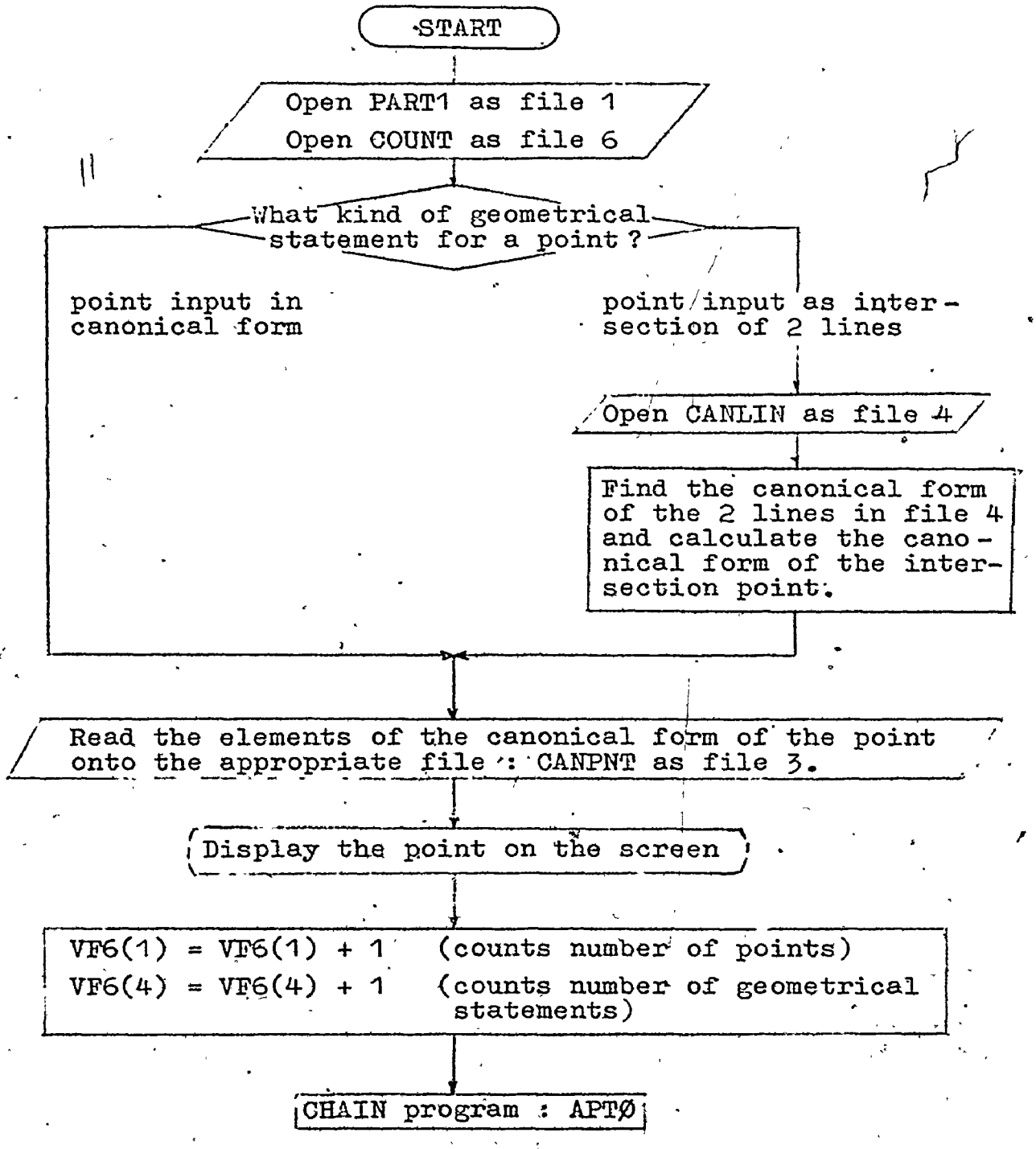


Flowchart APTØ

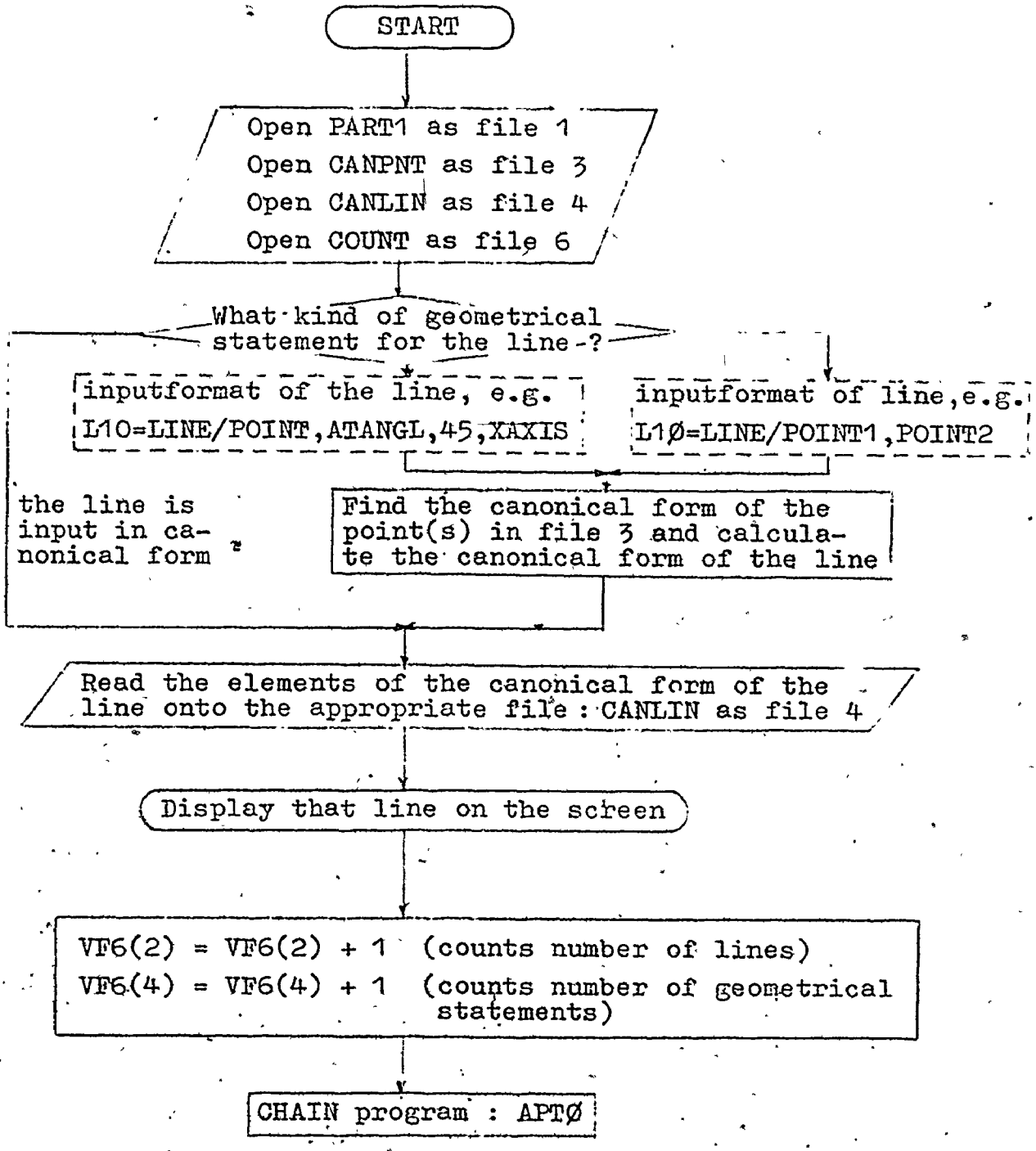



1
Print "Are you satisfied with the displayed geometry on CRT display"
Input A$
A$ = "Yes"
A$?
A$ = "No"
Print "Do you want to input the necessary part programming information with interactive graphics instead of teletype conversation"
Input A$
Input the new geometrical elements
Only erase the geometry that will be modified. The other geometrical elements remain on the screen.
4
A$?
A$ = "Yes"
A$ = "No"
CHAIN program : BEN1
kind of operation ?
milling
turning
CHAIN program : MIL111
CHAIN program : APTURN

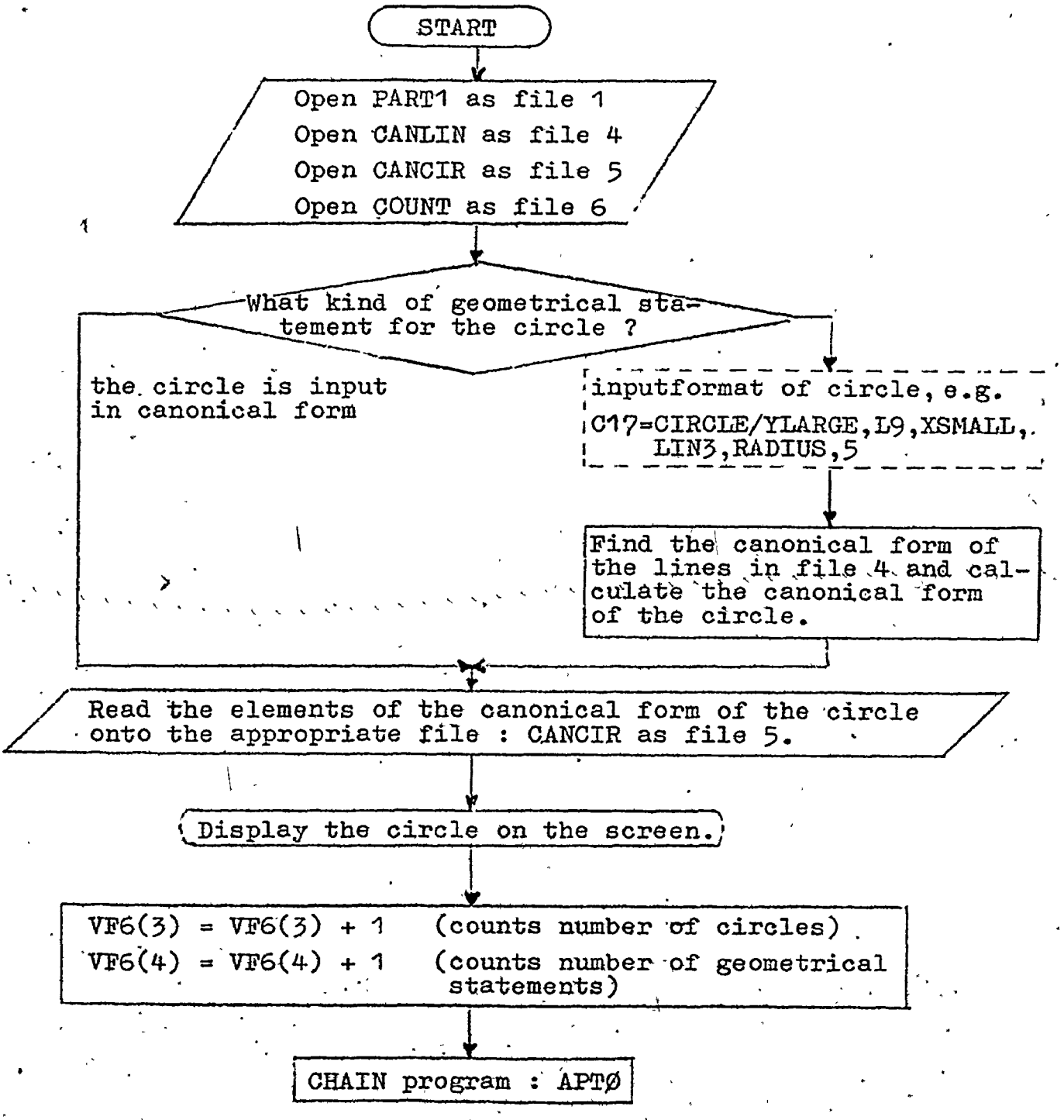
Flowchart POINTØ



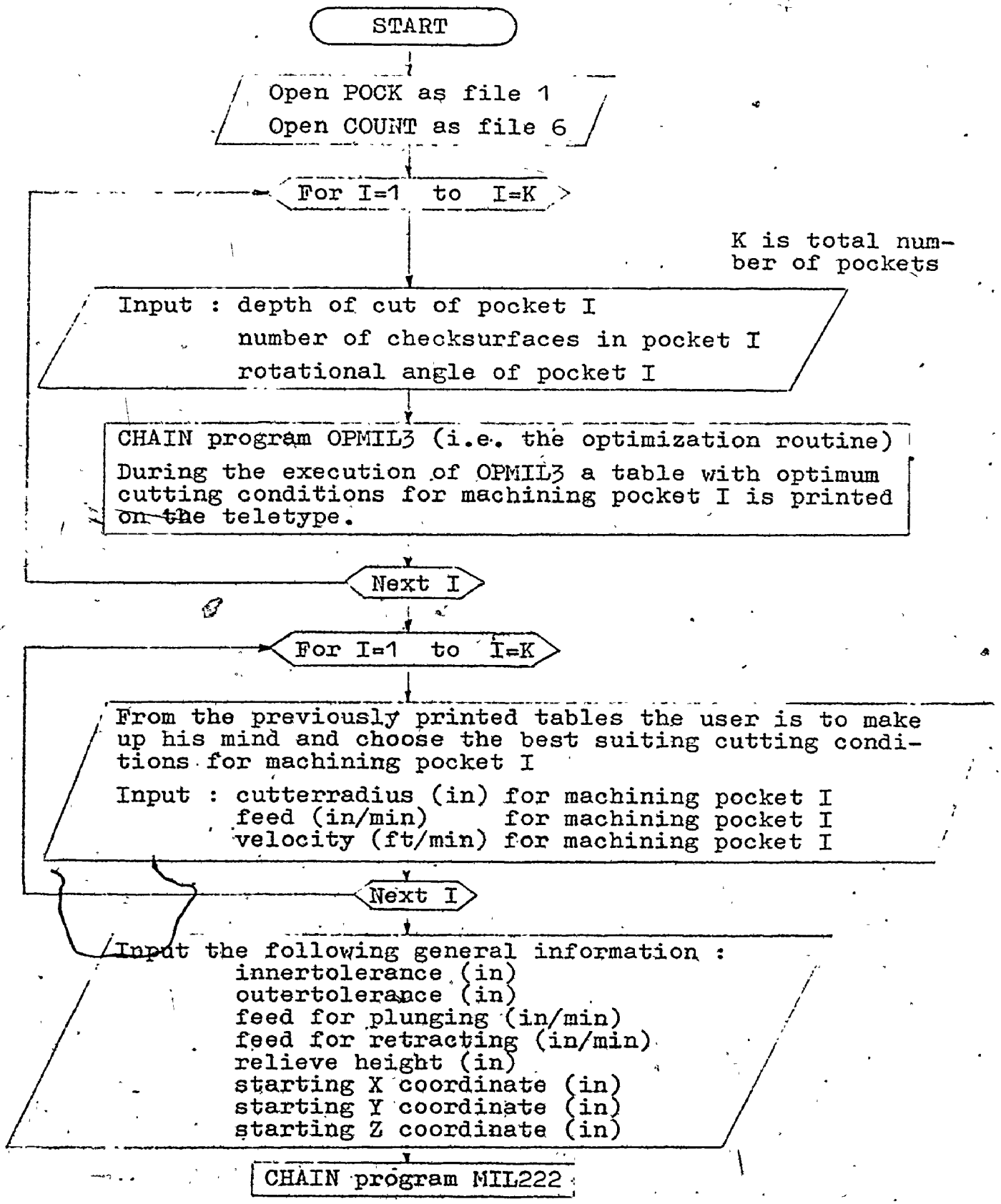
Flowchart LINEØ



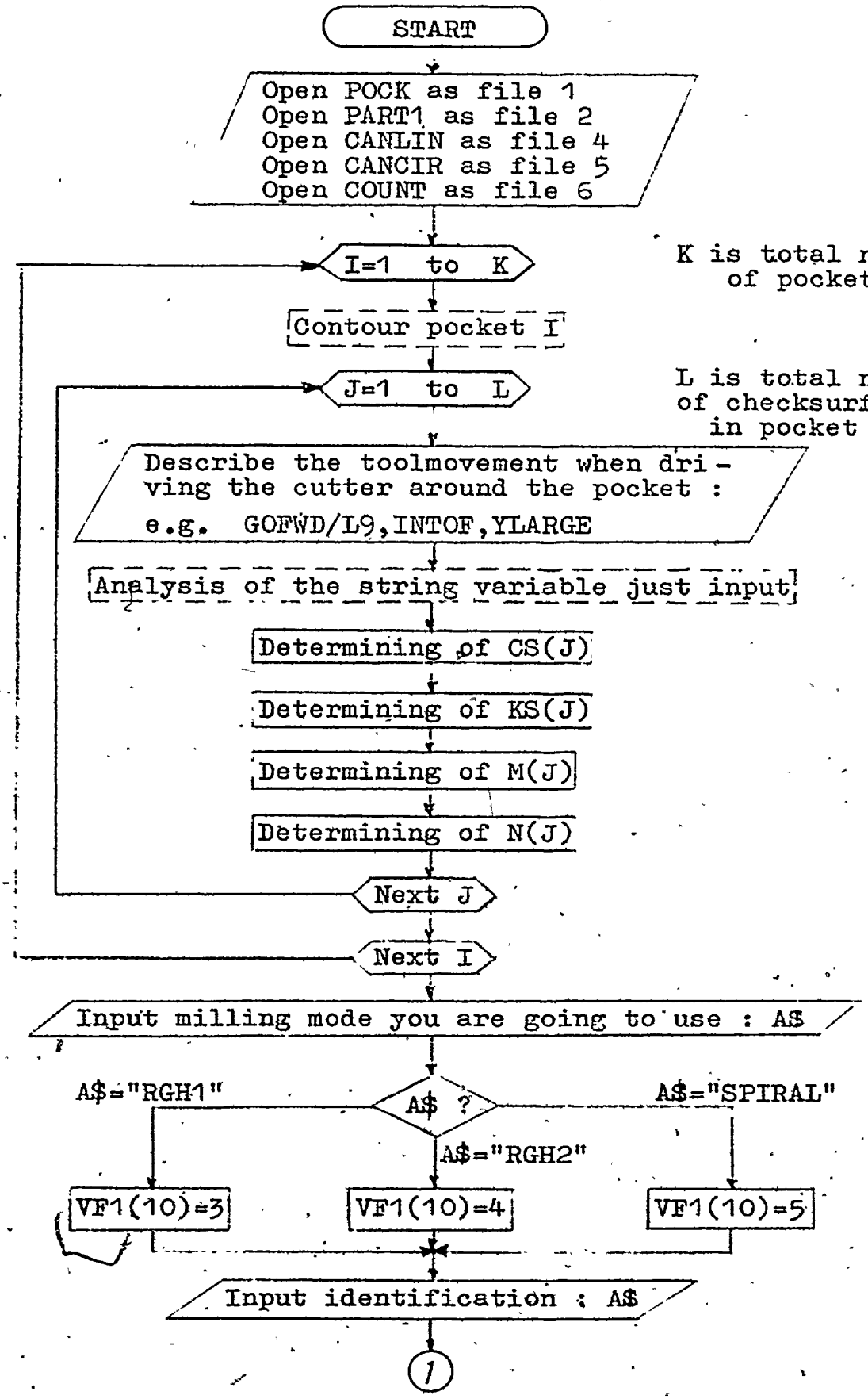
Flowchart CIRCLØ

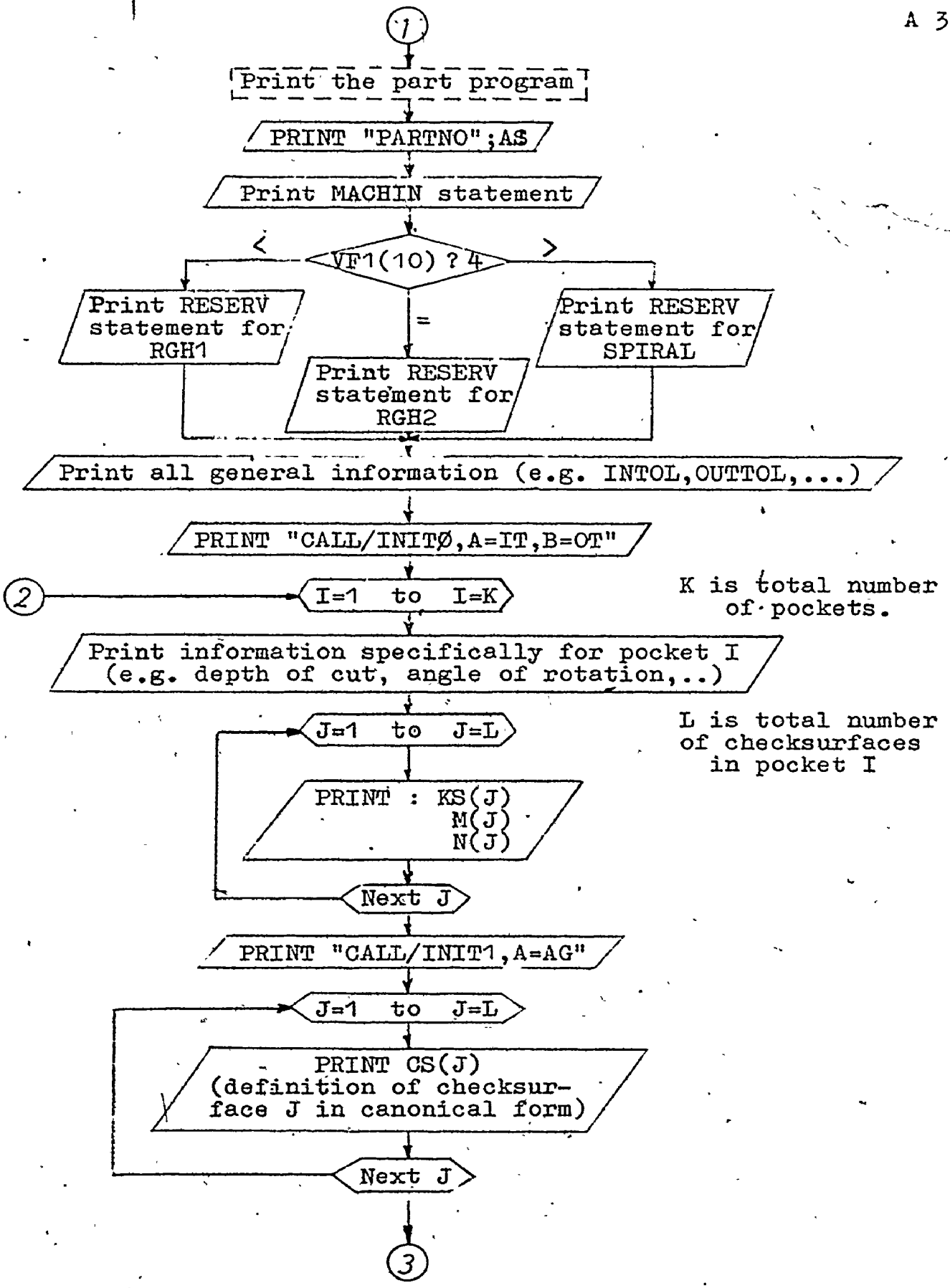


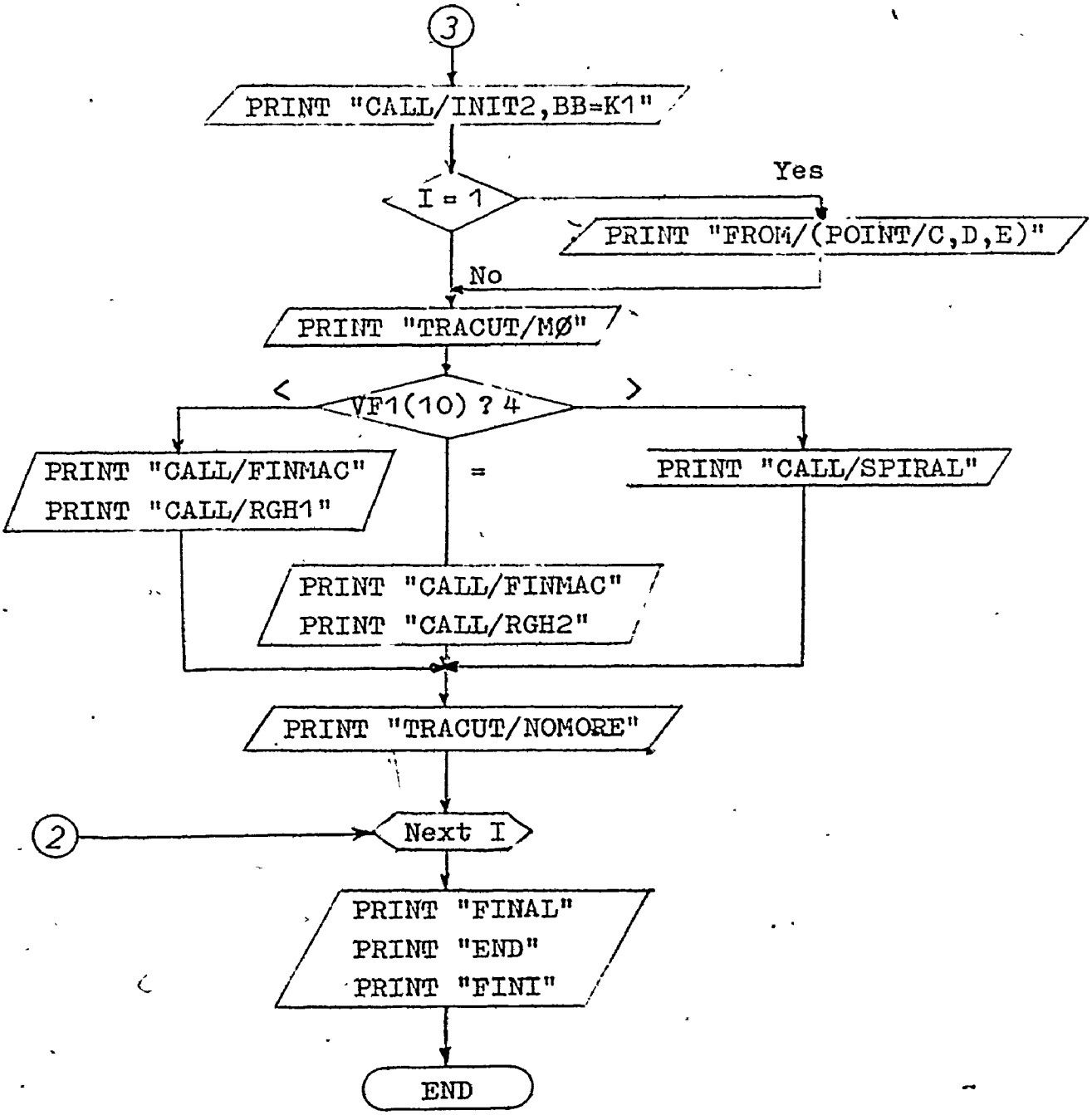
Flowchart MIL111



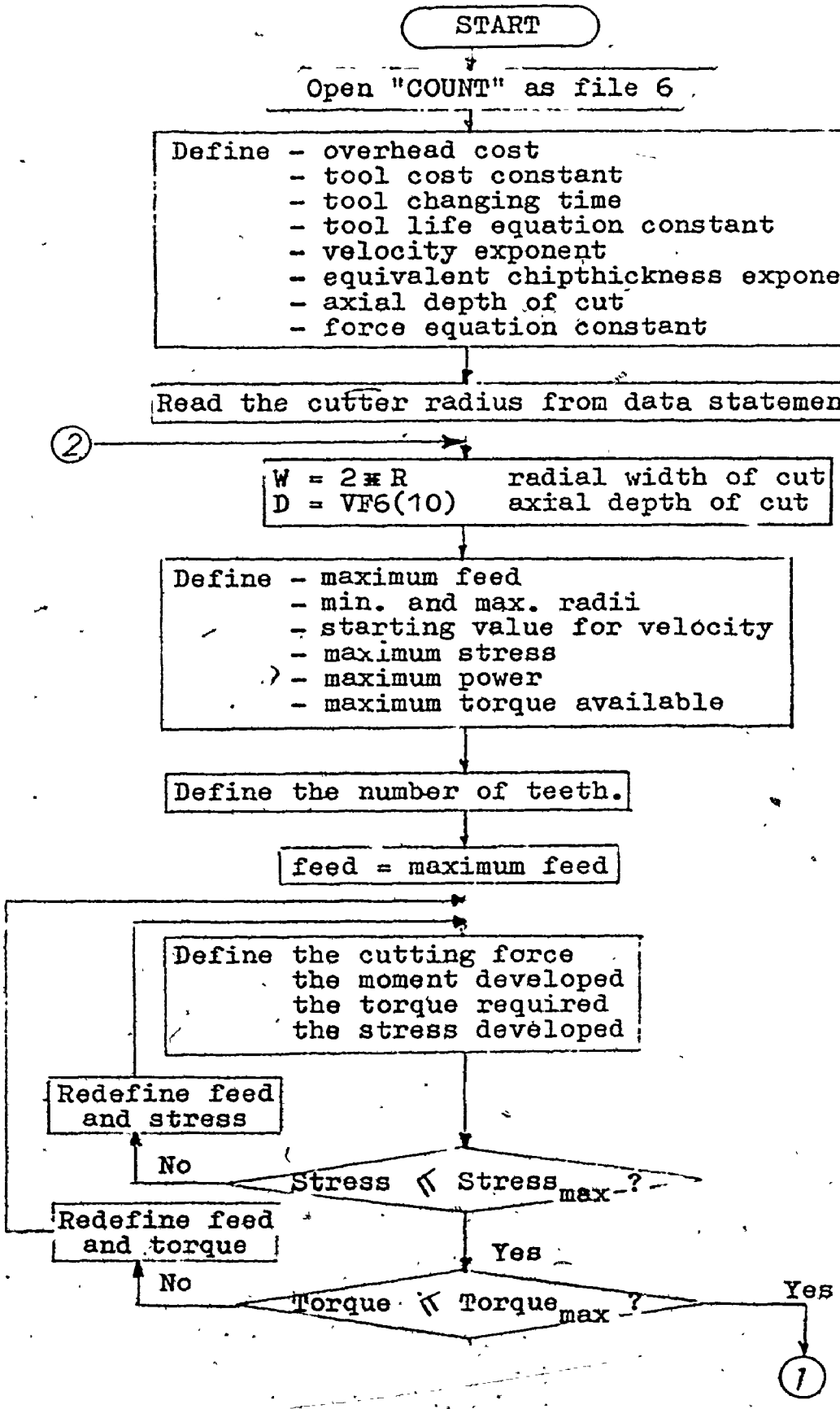
Flowchart MIL222

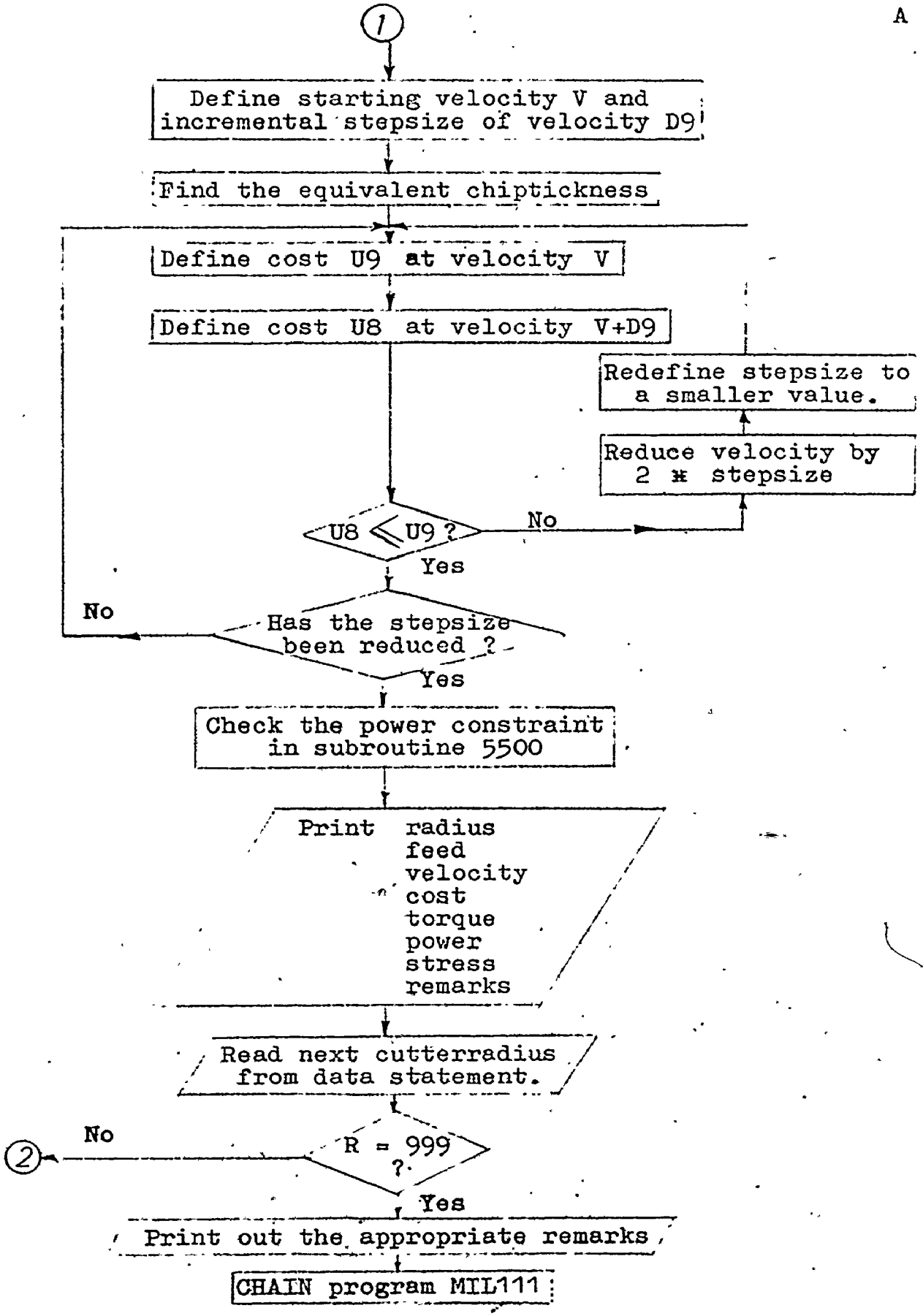



1
Print the part program
PRINT "PARTNO";A$
Print MACHIN statement
VF1(10) ? 4
<
=
>
Print RESERV statement for RGH1
Print RESERV statement for RGH2
Print RESERV statement for SPIRAL
Print all general information (e.g. INTOL,OUTTOL,...)
PRINT "CALL/INITØ,A=IT,B=OT"
2
I=1 to I=K
K is total number of pockets.
Print information specifically for pocket I (e.g. depth of cut, angle of rotation,..)
J=1 to J=L
L is total number of checksurfaces in pocket I
PRINT : KS(J) M(J) N(J)
Next J
PRINT "CALL/INIT1,A=AG"
J=1 to J=L
PRINT CS(J) (definition of checksurface J in canonical form)
Next J
3

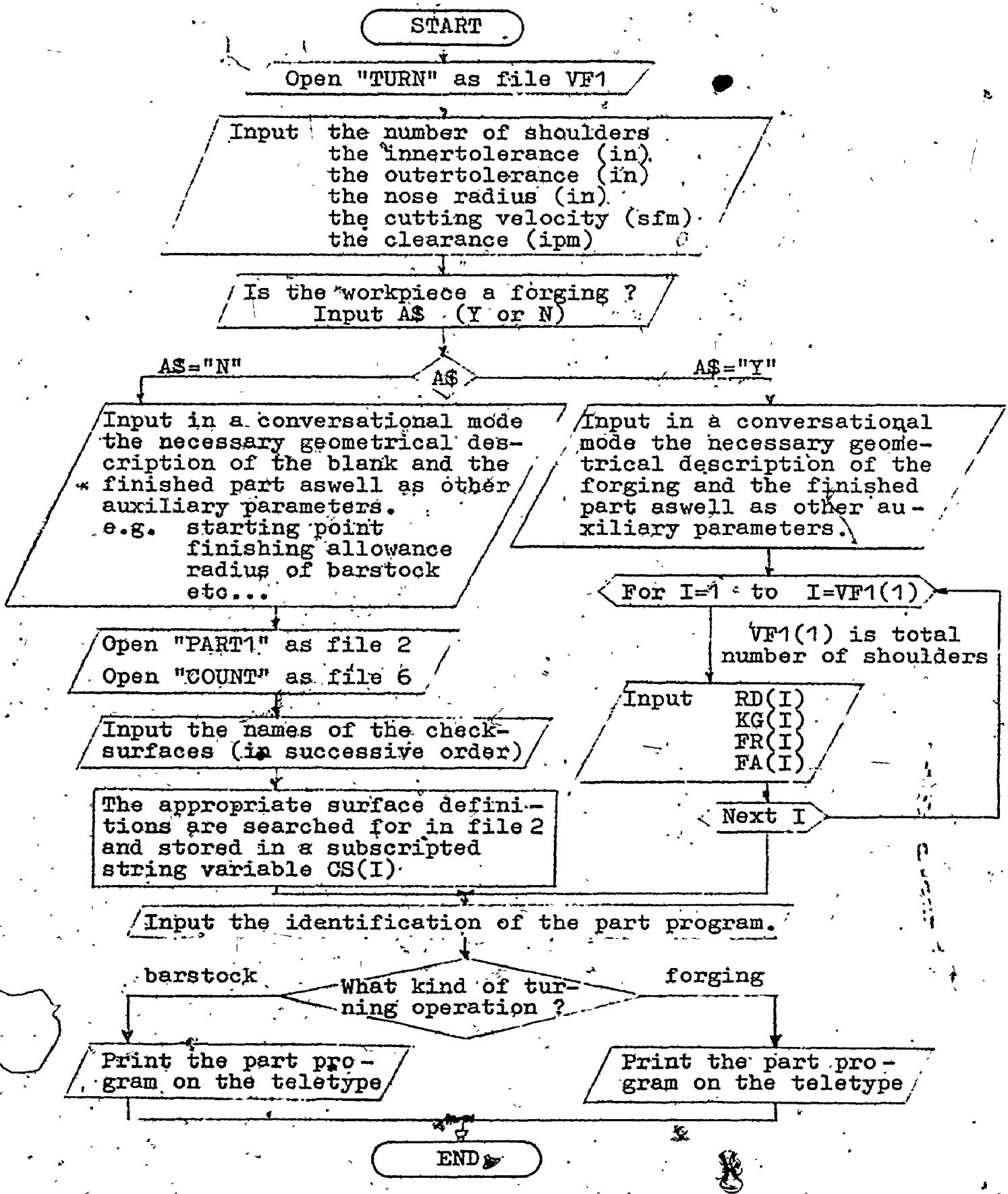

3
PRINT "CALL/INIT2,BB=K1"
I = 1
Yes
PRINT "FROM/(POINT/C,D,E)"
No
PRINT "TRACUT/MØ"
VF1(10) ? 4
<
=
>
PRINT "CALL/FINMAC"
PRINT "CALL/RGH1"
PRINT "CALL/FINMAC"
PRINT "CALL/RGH2"
PRINT "CALL/SPIRAL"
PRINT "TRACUT/NOMORE"
2
Next I
PRINT "FINAL"
PRINT "END"
PRINT "FINI"
END

Flowchart OPMIL3

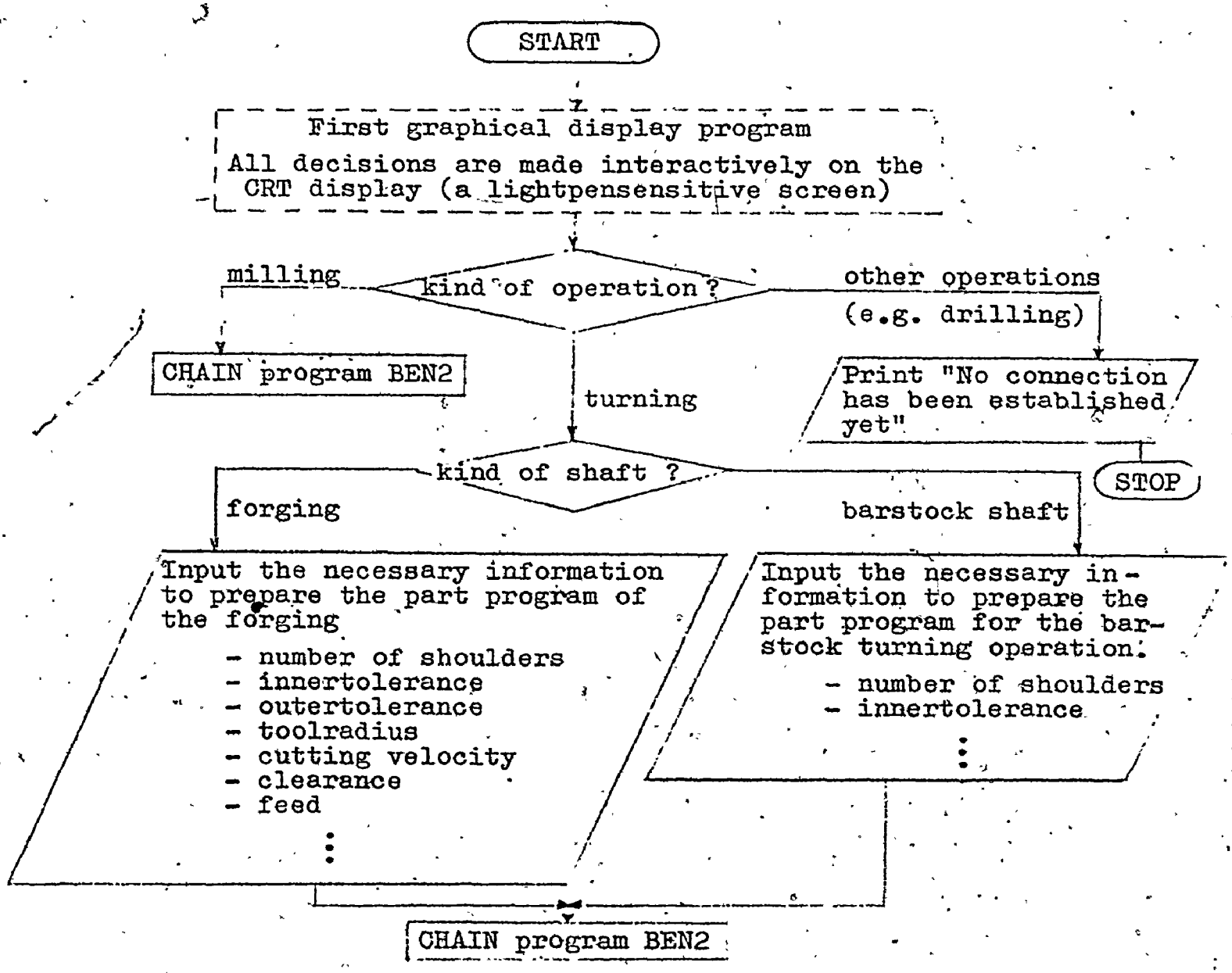



1
Define starting velocity V and incremental stepsize of velocity D9
Find the equivalent chiptickness
Define cost U9 at velocity V
Define cost U8 at velocity V+D9
Redefine stepsize to a smaller value.
Reduce velocity by 2 * stepsize
U8 ≤ U9 ?
No
Yes
No
Has the stepsize been reduced ?
Yes
Check the power constraint in subroutine 5500
Print radius feed velocity cost torque power stress remarks
Read next cutterradius from data statement.
No
2
R = 999 ?
Yes
Print out the appropriate remarks
CHAIN program MIL111

Flowchart APTURN.



Flowchart BEN1



Flowchart BEN2 and BEN3

START

Second graphical display program
The whole input of the OPMIL3 program (milling optimization program) appears on the screen and can be modified and updated.

Open "RADIUS" as file 1

With a lightpen hit on the screen, the user can pick out the radii of the endmills he has got available
Automatically these data are stored on the "RADIUS" file

Open "PARAM" as file 2
Open "PARAM1" as file 3

The following routine provides the user with the capability to input different inputvalues to the optimizationroutine. The values used in this work were stored in file "PARAM". When slightly touching a linesegment on the screen, these original values can be transfered to the "working" file "PARAM1". However, if the user wants to insert other parametervalues (e.g. a maximum power of 15 hp instead of 10 ph) he simple has to hit the "No" linesegment and type in "15". The contents of file "PARAM1" can be used as input to the optimization program OPMIL3.

Display : "Do you agree with the following values of

- overhead cost
- toolcost constant
- tool changing time
- toollife equation constant
- velocity exponent
- equivalent chipthickness exponent
- axial depth of cut exponent
- force equation constant
  and
- maximum feed per tooth
- minimum cutterradius
- maximum cutterradius
- thermal fatigue parameter exponent
  aswell as
- maximum stress
- maximum power
- maximum torque

CHAIN program BEN3

The first links have been made to implement the above mentioned files into the part program generation routines in general and the optimization program in particular.

END

Appendix 2 : Program Listings.

1. APT software.

1.1. Milling macros.

1.1.1. Common preparatory macros for the milling macros.

```
MAC0=MACRO/A                                                    0  1
$$      MACRO TO OBTAIN THE INTERSECTION POINTS IN APT LANGUAGE  0 10
$$        FROM THE BASIC PROGRAM OUTPUT                         0 11

$$      INPUT :                                                 0 14
            OLD KS(I)                                           0 15
            CS(I)                                               0 16
            K1                                                  0 17
            N(I) :                                              0 18
               XSMALL                                           0 19
               XLARGE                                           0 20
               YSMALL                                           0 21
               YLARGE                                           0 22

$$      OUTPUT :                                                0 23
$$          P(I)                                                0 24


            I=0                                                 0 25
0G)         I=I+1                                               0 30
            J=I-1                                               0 40
            IF (I-1) 0A,0B,0A                                   0 50
0B)         J=A                                                 0 50
0A)         IF (N(I)) 0E,0C,0D                                  0 70
0C)         CALL/MAC01,G=CS,D=J,E=I,F=P                         0 75
            JUMPTO/0F                                           0 80
0E)         IF (N(I)+1) 0K,0J,0J                                0 83
0K)         CALL/MAC02,AA=XSMALL,G=CS,D=J,E=I,F=P               0 86
            JUMPTO/0F                                           0 89
0J)         CALL/MAC02,AA=XLARGE,G=CS,D=J,E=I,F=P               0 92
            JUMPTO/0F                                           0 95
0D)         IF (N(I)-1) 0L,0L,0M                                0 98
0L)         CALL/MAC02,AA=YSMALL,G=CS,D=J,E=I,F=P               0 101
            JUMPTO/0F                                           0 103
0M)         CALL/MAC02,AA=YLARGE,G=CS,D=J,E=I,F=P               0 106
0F)         IF (I-A ) 0G,0H,0H                                  0 120
0H)         I=I
            TERMAC
            PRINT/0


            MAC01=MACRO/G,D,E,F                                 01  1
$$          ADDITIONAL MACRO TO MAC0 WHEN CS(J) AND CS(I) ARE BOTH LINES  01 10

            F(E)=POINT/INTOF,G(D),G(E)                          01 20
            TERMAC                                              01 30
            PRINT/0
```

```
         MACC2=MACRO/AA,G,D,E,F                                          02 1
$$       ADDITIONAL MACRO TO MACO WHEN CS(J) OR CS(I) (OR BOTH )         02 10
$$         ARE NOT LINES ( MORE THAN 1 POSSIBLE INTERSECTION )           02 11
         IF(M(E)) J2A,02B,02B                                            02 20
02B)     IF(KS(D))02C,J2D,02C                                            02 30
02D)     F(E)=POINT/AA,INTOF,G(D),G(E)                                   02 40
         JUMPTO/J2M                                                      02 50
02C)     F(E)=POINT/AA,INTOF,G(E),G(D)                                   02 60
         JUMPTO/J2M                                                      02 70
02A)     IF(KS(D)) J2E,02F,02E                                           02 80
02F)     EE=E          $$ CIRCLE                                         02 90
         DD=D          $$ LINE                                           02 100
         JUMPTO/J2L                                                      02 110
02E)     IF(KS(E))02G,02H,02G                                            02 120
02H)     EE=D          $$ CIRCLE                                         02 130
         DD=E          $$ LINE                                           02 140
02L)     OBTAIN,CIRCLE/G(EE),,,,,,,ER                                    02 150
         JUMPTO/02K                                                      02 160
02G)     OBTAIN,CIRCLE/G(D),,,,,,,RAD1                                   02 170
         OBTAIN,CIRCLE/G(E),,,,,,,RAD2                                   02 180
         IF(RAD1-RAD2)02I,02I,02J                                        02 190
02I)     EE=D          $$ SMALLEST CIRCLE                                02 200
         DD=E                                                            02 210
         ER=RAD1                                                         02 220
         JUMPTO/02K                                                      02 230
02J)     EE=E          $$ SMALLEST CIRCLE                                02 240
         DD=D                                                            02 250
         ER=RAD2                                                         02 260
02K)     G(EE)=CIRCLE/G(EE),CANON,,,,,,,(ER+.0000000001)                 02 270
         F(E)=POINT/AA,INTOF,G(DD),G(EE)                                 02 280
         G(EE)=CIRCLE/G(EE),CANON,,,,,,,ER                               02 290
02M)     TERMAC                                                          02 300
         PRINT/0
```

```
         MAC1=MACRO/A,B                                                  1 1
$$       LOOP TO CALCULATE THE COORDINATES OF THE INTERSECTION POINTS    1 10

$$       INPUT :                                                         1 13
$$           P(I)                                                        1 14
$$           K1                                                          1 15
$$
$$       OUTPUT :                                                        1 16
$$           X(I),Y(I)                                                   1 17
$$           K2                                                          1 18

         I=1
1A)      OBTAIN,POINT/A(I),X(I),Y(I)                                     1 20
         I=I+1                                                           1 30
         IF(I-B ) 1A,1A,1B                                               1 40

$$       LOOP TO CALCULATE THE HIGHEST INTERSECTION POINT                1 50

1B)      I=1                                                             1 70
         K2=0                                                            1 80
1E)      IF(Y(I+1)-Y(I)) 1C,1D,1D                                        1 90
1D)      K2=Y(I+1)                                                       1 100
         I=I+1                                                           1 110
         IF(I-B ) 1E,1C,1C                                               1 120
1C)      K2=I                                                            1 130
         TERMAC                                                          1 140
         PPINT/0
```

```
$$      MACRO TO OBTAIN THE TANGENTS TO THE CHECKSURFACES IN THE          2 10
$$      INTERSECTION POINTS AND TO CALCULATE THE ANGLE THE TANGENTS       2 11
$$        MAKE WITH THE POSITIVE X-AXIS AND TO DETERMINE APPROXIMATELY    2 12
$$        THE RELATIVE ANGLE BETWEEN THE CHECKSURFACES                    2 13

$$      W(I)=-1    RELATIVE ANGLE<180
$$      W(I)=0     RELATIVE ANGLE=180
$$      W(I)=+1    RELATIVE ANGLE>180


$$      INPUT :
                K1,K2
                CS(I)
                KS(I) CLD
                M(I)

        NOTE : M(I) :
                  GOBACK  -2
                  GOFWD   -1
                  GOLEFT  +1
                  GORIGHT +2

        OUTPUT :
                 T1(I)
                 T2(I)
                 A1(I)
                 A2(I)
                 W (I)


        I=1                                                               2 20
        J=2
2C)     IF(KS(I)) 2B,2A,2B                                                2 30

$$      THE CHECKSURFACE I IS A LINE                                      2 35

2A)     T2(I)=LINE/CS(I),CANON,,,,                                        2 40
        CALL/MAC21,DUM1=T2,DUM2=A2,DUM3=I                                 2 50
        T1( J )=LINE/CS(I),CANON,,,,                                      2 60
        CALL/MAC21,DUM1=T1,DUM2=A1,DUM3=J                                 2 70
        I=I+1                                                             2 80
        J=I+1
        IF(I-A ) 2C,2L,2D                                                 2 90

2L)     J=1                                                               2 92
        JUMPTO/2C                                                         2 93

$$      THE CHECKSURFACE I IS A CIRCLE                                    2 95

2B)     T2(I)=LINE/P(I),LEFT,TANTO,CS(I)                                  2 100
        CALL/MAC21,DUM1=T2,DUM2=A2,DUM3=I                                 2 110
        T1( J )=LINE/P( J ),LEFT,TANTO,CS(I)                              2 120
        CALL/MAC21,DUM1=T1,DUM2=A1,DUM3=J                                 2 130
        I=I+1                                                             2 140
        J=I+1
        IF(I-A ) 2C,2L,2D                                                 2 150

$$  FINDING OUT WHETHER OR NOT THE RELATIVE ANGLE IS > OR < 180          2 160

2D)     I=0                                                               2 170
2I)     I=I+1                                                             2 180
        IF(M(I)-1)2E,2F,2G                                                2 190
2E)     W(I)=0                                                            2 200
        IF (M(I)+1) 2K,2L,2H                                              2 203
2K)     M(I)=2                                                            2 206
        W(I)=-1                                                           2 208
        JUMPTO/2H                                                         2 209

$$      IF M(I)=-2  THEN  M(I)=2                                          2 212
$$      IN REALITY GOBACK BECOMES GORGT                                   2 215
$$      ( THIS MEANS A CLITTERRADIUS R>0 )                                2 218

2F)     W(I)=1                                                            2 220
        JUMPTO/2H                                                         2 230
2G)     W(I)=-1                                                           2 240
2H)     IF(I-A)2I,2J,2J                                                   2 250
2J)     TERMAC                                                            2 260
        PRINT/0
```

```
      MAC21=MACRO/DUM1,DUM2,DUM3                                          21 1
$$    ADDITIONAL MACRO TO MAC2 USED TO OBTAIN THE ANGLE THAT THE
$$      TANGENT MAKES WITH THE POSITIVE X-AXIS
      OBTAIN,LINE/DUM1(DUM3),AO,BO,,                                      21 10
      IF(BO) 21A,21B,21A                                                  21 20
21B)  DUM2(DUM3)=-90                                                      21 30
      JUMPTO/21C                                                          21 40
21A)  DUM2(DUM3)=ATANF(-AO/BO)                                            21 50
21C)  IF(ABSF(DUM2(DUM3))-.1) 21D,21D,21E                                 21 60
21D)  DUM2(DUM3)=0                                                        21 70
21E)  TERMAC                                                              21 80
      PRINT/0
```

```
      MAC4=MACRO/A                                                        4 1
$$    DETERMINING SURFACES WHICH ARE PARALLEL OR CONCENTRIC TO THE        4 10
$$      CHECKSURFACES WITH AN OFFSET R ( INWARDS THE POCKET )             4 11
$$      CALCULATION OF CA(I)                                              4 12
      I=0                                                                 4 20
4N)   I=I+1                                                               4 30
      J=I+1                                                               4 40
      IF (I-A ) 4A,4B,4C                                                  4 50
4B)   J=1                                                                 4 60
4A)   IF (KS(I)) 4D,4E,4F                                                 4 70
4D)   OBTAIN,CIRCLE/CS(I),,,,,,,RAD      $$ CONCAVE CIRCLE                4 80
      CA(I)=CIRCLE/CS(I),CANON,,,,,,,(RAD+R)                              4 90
      JUMPTO/4N                                                           4 100
4F)   OBTAIN,CIRCLE/CS(I),,,,,,,RAD      $$ CONVEX CIRCLE                 4 110
      CA(I)=CIRCLE/CS(I),CANON,,,,,,,(RAD-R)                              4 120
      JUMPTO/4N                                                           4 130
4E)   IF(A2(I)) 4G,4H,4G                 $$ LINE                          4 140
4G)   IF (I-K2) 4J,4K,4K                 $$ SECOND TANGENT IS             4 150
$$                                          NONZERO                      4 151
4J)   CA(I)=LINE/PARLEL,CS(I),XLARGE,R      $$ LEFT SIDE                  4 160
      JUMPTO/4N                                                           4 170
4K)   CA(I)=LINE/PARLEL,CS(I),XSMALL,R      $$ RIGHT SIDE                 4 180
      JUMPTO/4N                                                           4 190
4H)   IF (X(J)-X(I)) 4L,4L,4M              $$ SECOND TANGENT IS          4 200
$$                                          ZERO                         4 201
4L)   CA(I)=LINE/PARLEL,CS(I),YLARGE,R      $$ FOLLOWING POINT            4 210
$$                                             IS XSMALL                 4 211
      JUMPTO/4N                                                           4 220
4M)   CA(I)=LINE/PARLEL,CS(I),YSMALL,R      $$ FOLLOWING POINT            4 230
$$                                             IS XLARGE                 4 231
      JUMPTO/4N                                                           4 240
4C)   I=I                                                                 4 250
      TERMAC                                                              4 260
      PRINT/0
```

```
$$      MACRO TO OBTAIN THE Y-LIMITS OF THE INTERSECTION POINTS          5 10

$$      ANGLE BETWEEN CHECKSURFACES < 180 : NO PROBLEMS                  5 20
$$              W(I)=0  AND  W(I)=-1                                     5 21
$$      ANGLE BETWEEN CHECKSURFACES > 180 : PROBLEMS                     5 30
$$              W(I)=1                                                   5 31

        I=0                                                              5 50
5Y)     I=I+1                                                            5 60
        J=I-1                                                            5 70
        IF(I-A) 5CC,5CC,5EE                                              5 75
5CC)    IF (I-1) 5A,5B,5A                                                5 80
5B)     J=A                                                              5 90
5A)     IF (W(I)) 5AA,5AA,5Z                                             5 100
5AA)    IF(N(I)) 5C,5D,5E                                                5 130
5D)     CALL/MAC01,G=CA,C=J,E=I,F=Q                                      5 140
        JUMPTO/5J                                                        5 150
5C)     IF(N(I)+1) 5F,5G,5G                                              5 160
5F)     CALL/MAC02,AA=XSMALL,G=CA,D=J,E=I,F=Q                            5 170
        JUMPTO/5J                                                        5 180
5G)     CALL/MAC02,AA=XLARGE,G=CA,D=J,E=I,F=Q                            5 190
        JUMPTO/5J                                                        5 200
5E)     IF(N(I)-1) 5H,5H,5I                                              5 210
5H)     CALL/MAC02,AA=YSMALL,G=CA,D=J,E=I,F=Q                            5 220
        JUMPTO/5J                                                        5 230
5I)     CALL/MAC02,AA=YLARGE,G=CA,D=J,E=I,F=Q                            5 240
5J)     OBTAIN,POINT/Q(I),X1(I),Y1(I)                                    5 250
        Y2(I)=Y1(I)                                                      5 260
        X2(I)=X1(I)                                                      5 270
        JUMPTO/5Y                                                        5 280

5Z)     IF (I-K2) 5K,5K,5L              $$  PROBLEMS                     5 290

5K)     IF (A1(I)) 5N,5M,5N       $$  LEFT SIDE OF POCKET                5 300
5N)     CALL/MAC51,CUM1=T1,DUM5=XLARGE,DUM3=X1,DUM4=Y1                   5 310
        JUMPTO/5P                                                        5 320
5M)     CALL/MAC51,CUM1=T1,DUM5=YSMALL,DUM3=X1,DUM4=Y1                   5 330
5P)     IF (A2(I)) 5Q,5R,5Q                                              5 340
5Q)     CALL/MAC51,CUM1=T2,DUM5=XLARGE,DUM3=X2,DUM4=Y2                   5 350
        JUMPTO/5Y                                                        5 360
5R)     CALL/MAC51,CUM1=T2,DUM5=YLARGE,DUM3=X2,DUM4=Y2                   5 370
        JUMPTO/5Y                                                        5 380

5L)     IF (A1(I)) 5T,5U,5T       $$  RIGHT SIDE OF POCKET               5 390
5T)     CALL/MAC51,CUM1=T1,DUM5=XSMALL,DUM3=X1,DUM4=Y1                   5 400
        JUMPTO/5V                                                        5 410
5U)     CALL/MAC51,CUM1=T1,DUM5=YLARGE,DUM3=X1,DUM4=Y1                   5 420
5V)     IF (A2(I)) 5W,5X,5W                                              5 430
5W)     CALL/MAC51,CUM1=T2,DUM5=XSMALL,DUM3=X2,DUM4=Y2                   5 440
        JUMPTO/5Y                                                        5 450
5X)     CALL/MAC51,CUM1=T2,DUM5=YSMALL,DUM3=X2,DUM4=Y2                   5 460
        JUMPTO/5Y                                                        5 465

5EE)    TERMAC                                                           5 480


        MAC51=MACRO/CUM1,DUM5,DUM3,DUM4                                  51 1

$$      ADDITIONAL MACRO TO MAC5                                         51 10

$$      A CIRCLE WITH A GIVEN RADIUS, TANGENT TO A LINE, AND PASSING     51 11
$$      THROUGH A POINT                                                  51 12


        OBTAIN,CIRCLE/(CIRCLE/TANTO,DUM1(I),DUM5,P(I),RADIUS,F),    $    51 20
        DUM3(I),DUM4(I)                                                  51 21
        TERMAC                                                           51 30
        PRINT/0
```

```
        INIT0=MACRO/A,B                                                  I0 10
$$      FIRST INITIALIZATION MACRO                                       I0 15
        INTOL/A                                                          I0 20
        OUTTOL/B                                                         I0 30
        CLPRNT                                                           I0 40
        SPINDL/100                                                       I0 60
        COOLNT/ON                                                        I0 70
        PRINT/3,ALL                                                      I0 90
        TERMAC                                                           I0 100
        PRINT/0


        INIT1=MACRO/A                                                    I1 1
$$      SECOND INITIALIZATION MACRO                                      I1 3
$$      THE USE OF REFSYS WILL TRANSFER (I.E. ROTATE) THE COORDINATES    I1 4
$$      OF THE UPCOMING POCKETSURFACES INTO A COORDINATE SYSTEM SUCH     I1 5
$$      THAT THE CUTTERDIRECTION WILL BE PARALLEL TO THE X-AXIS          I1 6
        CUTTER/(2*R)                                                     I1 8
        FEDRAT/FD                                                        I1 9
        LX=LINE/CANON,0,1,0,0                                            I1 10
        LY=LINE/CANON,1,0,0,0                                            I1 20
        M0=MATRIX/XYPOT,A                                                I1 40
        REFSYS/(M1=MATRIX/INVERS,M0)                                     I1 50
        TERMAC                                                           I1 60
        PRINT/0


        INIT2=MACRO/BB                                                   I2 1
$$      THIRD INITIALIZATION MACRO                                       I2 3
$$      PREPARATORY MACROS ARE CALLED IN ORDER TO EXECUTE GEOMETRICAL    I2 5
$$      CALCULATIONS                                                     I2 6
        REFSYS/ICMORE                                                    I2 10
        CALL/MAC0,A=BB                                                   I2 40
        CALL/MAC1,A=P,B=BB                                               I2 50
        CALL/MAC2,A=BB                                                   I2 60
        CALL/MAC4,A=BB                                                   I2 80
        CALL/MAC5,A=BB                                                   I2 90
        TERMAC                                                           I2 150
        PRINT/0


        FINAL=MACRO/
$$      LAST MACRO TO TURN OFF THE COOLANT AND TO STOP THE SPINDLE
        COOLNT/OFF
        SPINDL/OFF
        RAPID
        GO TO / (PCINT/C,D,E)
        TERMAC
        PRINT/0
```

1.1.2. The main pocketing macros.

1.1.2.1. FINMAC (semiroughing macro)

```
FINMAC=MACRO/
$$      THIS MACRO CAN BE USED FOR SEMI-ROUGHING AND FOR A FINISHING    F 20
$$        PASS                                                          F 21
        RAPID                                                           F 30
        Q(1)=POINT/Q(1),CANON,,,A                                       F 50
        GO TO/Q(1)                                                      F 60
        GODLTA/0,0,-(B+A),FDW                                           F 70
        FEDRAT/FD                                                       F 80
        PSIS/(PLANE/0,0,1,-B)                                           F 90
        ONTCUT                                                          F 100
        GODLTA/.1,0,0                                                   F 110
        INDIRV/-1,0,0                                                   F 120
        GO/ON,CA(1),(PLANE/0,0,1,-B),  ON ,(LINE/PARLEL,LX,YLARGE,$     F 130
        Y1(1))                                                          F 140
        CUT                                                             F 150

$$      END OF ORIENTATION OF THE CUTTER AND START OF ACTUAL CUTTING    F 155
$$      MOTIONS                                                         F 156

PPRINT  **********************************************************************
PPRINT
PPRINT  BEGINNING OF SEMIROUGHING CUT
PPRINT
PPRINT  **********************************************************************
        J=0                                                             F 160
F15)    J=J+1                                                           F 170
        I=J+1                                                           F 180
        IF(J-K1) F3,F4,F5                                               F 190
F3)     IF(M(I)-1) F7,F6,F8                                             F 200
F6)     IF(M( J )-1) F9,F9,F10                                          F 210
F9)     CALL/FIN1,AA=GOFWD                                              F 220
        JUMPTO/F15                                                      F 230
F10)    CALL/FIN1,AA=GORGT                                              F 240
        JUMPTO/F15                                                      F 250
F7)     IF(M( J )-1) F11,F11,F12                                        F 260
F11)    TLON,GOFWD/CA( J ),TANTO,CA(I)                                  F 270
        JUMPTO/F15                                                      F 280
F12)    TLON,GORGT/CA( J ),TANTO,CA(I)                                  F 290
        JUMPTO/F15                                                      F 300
F8)     IF(M( J )-1) F13,F13,F14                                        F 310
F13)    CALL/FIN2,AA=GOFWD                                              F 320
        JUMPTO/F15                                                      F 330
F14)    CALL/FIN2,AA=GORGT                                              F 340
        JUMPTO/F15                                                      F 350
F4)     I=1                                                             F 360
        JUMPTO/F3                                                       F 370
F5)     TERMAC                                                          F 380
```

```
        FIN1=MACRO/AA                                                   F1 10
$$      ADDITIONAL MACRO TO FINMAC                                      F1 15
        TLON,AA/CA( J ),TANTO,(CIRCLE/CENTER,P(I),RADIUS,R)             F1 20
        TLON,GOFWD/(CIRCLE/CENTER,P(I),RADIUS,R),TANTO,CA(I)            F1 30
        TERMAC                                                          F1 40
        PRINT/0
```

```
        FIN2=MACRO/AA                                                   F2 10
$$      ADDITIONAL MACRO TO FINMAC                                      F2 20
        IF (KS(J)) FF4,FF3,FF4                                          F2 30
FF3)    IF (A1(I))FF2,FF1,FF2                                           F2 40
FF4)    IF (N(I)) FF1,FF1,FF2                                           F2 50
FF1)    TLON,AA/CA(J),ON,(LINE/PARLEL,LY,XLARGE,X1(I))                  F2 70
        JUMPTO/FF7                                                      F2 80
FF2)    TLON,AA/CA(J),ON,(LINE/PARLEL,LX,YLARGE,Y1(I))                  F2 120
FF7)    ONTCUT                                                          F2 130
        IF (I-K2) FF8,FF9,FF9                                           F2 140
FF8)    IF (A2(I)) FF10,FF11,FF11                                       F2 150
FF10)   GODLTA/(.01*COSF(A2(I)+90)),(.01*SINF(A2(I)+90)),0              F2 160
        AAA=-1*COSF(A2(I)+90)                                           F2 170
        BBB=-1*SINF(A2(I)+90)                                           F2 174
        INDIRV/(VECTOR/AAA,BBB,0)                                       F2 178
        JUMPTO/FF20                                                     F2 180
FF11)   GODLTA/(.01*COSF(A2(I)-90)),(.01*SINF(A2(I)-90)),0              F2 190
        AAA=-1*COSF(A2(I)-90)                                           F2 200
        BBB=-1*SINF(A2(I)-90)                                           F2 204
        INDIRV/(VECTOR/AAA,BBB,0)                                       F2 208
        JUMPTO/FF20                                                     F2 210
FF9)    IF (A2(I)) FF11,FF10,FF10                                       F2 220
FF20)   GO/ON,CA(I),(PLANE/0,0,1,-3),ON,CA(J)                           F2 230
        CUT                                                             F2 240
        TERMAC                                                          F2 250
        PRINT/0
```

1.1.2.2. RGH1 (unidirectional roughing macro)

```
        RGH1=MACRO/                                                          R1  1

$$      RGH1 IS A ROUGHING MACRO                                             R1  2
$$        THE CUTTER ROUGHS OUT THE POCKET WHILE PERFORMING SEQUENTIAL       R1  3
$$        MOTIONS FROM THE LEFT SIDE TO THE RIGHT SIDE OF THE POCKET         R1  4
$$        ( ALWAYS IN THE DIRECTION OF THE POSITIVE X-AXIS )                 R1  5
$$        IN BETWEEN THE CUTTING MOTIONS, THE CUTTER IS LIFTED TO THE        R1  7
$$        HEIGHT *A* ABOVE THE SURFACE, IN ORDER TO MOVE RAPIDLY TO          R1  8
$$        ITS NEW STARTING POSITION                                          R1  9

$$      CALCULATING THE NUMBER OF PASSES                                     R1  10
        YA=Y1(1)                                                             R1  20
        K3=0                                                                 R1  30
R15)    K3=K3+2*R                                                            R1  40
        IF (K3-Y1(K2)+Y1(1)) R15,R16,R16                                     R1  50
R16)    K3=K3/(2*R)                                                          R1  60

$$      CALCULATING THE POINTS ON THE LEFT HAND SIDE OF THE POCKET           R1  70
$$      I.E. U(K4+1)                                                         R1  71

R18)    U(1)=POINT/X1(1),Y1(1),0                                             R1  75
        K4=1                                                                 R1  80
        KL=2                                                                 R1  90
R14)    YA=YA+2*R                                                            R1  100
R10)    IF (YA-Y1(KL)) R1,R2,R3                                              R1  110
R1)     IF (KS(KL-1)) R4,R5,R6                                               R1  120
R4)     U(K4+1)=POINT/XLARGE,INTOF,(LINE/PARLEL,LX,YLARGE,YA),CA(KL-1)       R1  130
        JUMPTO/R11                                                           R1  140
R5)     U(K4+1)=POINT/INTOF,(LINE/PARLEL,LX,YLARGE,YA),CA(KL-1)              R1  150
        JUMPTO/R11                                                           R1  160
R6)     U(K4+1)=POINT/XSMALL,INTOF,(LINE/PARLEL,LX,YLARGE,YA),CA(KL-1)       R1  170
        JUMPTO/R11                                                           R1  180
R2)     U(K4+1)=POINT/X1(KL),Y1(KL),0                                        R1  190
        JUMPTO/R11                                                           R1  200
R3)     IF (YA-Y2(KL)) R7,R8,R9                                              R1  210
R7)     U(K4+1)=POINT/XLARGE,INTOF,(LINE/PARLEL,LX,YLARGE,YA),       $       R1  220
        (CIRCLE/CENTER,P(KL),RADIUS,R)                                       R1  221
        JUMPTO/R11                                                           R1  230
R8)     U(K4+1)=POINT/X2(KL),Y2(KL),0                                        R1  240
        JUMPTO/R11                                                           R1  250
R9)     KL=KL+1                                                              R1  260
        IF (KL-K2) R10,R10,R38                                               R1  265
R11)    K4=K4+1                                                              R1  270
        IF(K4-K3) R14,R12,R12                                                R1  280

$$      CALCULATING THE POINTS ON THE RIGHT HAND SIDE OF THE POCKET          R1  290
$$      I.E. V(K4+1)                                                         R1  291

R12)    V(1)=POINT/X2(K1),Y2(K1),0                                           R1  300
        YA=Y1(1)                                                             R1  305
        K4=1                                                                 R1  310
        KR=K1                                                                R1  320
R20)    YA=YA+2*R                                                            R1  330
R21)    IF (YA-Y2(KR)) R22,R23,R24                                           R1  340
R22)    IF (KS(KR)) R25,R26,R27                                              R1  350
R25)    V(K4+1)=POINT/XSMALL,INTOF,(LINE/PARLEL,LX,YLARGE,YA),CA(KR)         R1  360
```

```
        JUMPTO/R31
R26)    V(K4+1)=POINT/INTOF,(LINE/PARLEL,LX,YLARGE,YA),CA(KR)
        JUMPTO/R31
R27)    V(K4+1)=POINT/XLARGE,INTOF,(LINE/PARLEL,LX,YLARGE,YA),CA(KR)
        JUMPTO/R31
R23)    V(K4+1)=POINT/X2(KR),Y2(KR),0
        JUMPTO/R31
R24)    IF (YA-Y1(KR)) R28,R29,R31
R28)    V(K4+1)=POINT/XSMALL,INTOF,(LINE/PARLEL,LX,YLARGE,YA),
        (CIRCLE/CENTER,P(KR),RADIUS,R)
        JUMPTO/R31
R29)    V(K4+1)=POINT/X1(KR),Y1(KR),0
        JUMPTO/R31
R30)    KR=KR-1
        IF (KR-K2) R33,R21,R21
R31)    K4=K4+1
        IF (K4-K3) R20,R33,R33

$$      ACTUAL ACTION STATEMENTS

R33)    GOLTA/J,J,(B+A),FUP
PPRINT  ***************************************************************
PPRINT
PPRINT  END OF SEMIROUGHING CUT
PPRINT
PPRINT  ***************************************************************
        K4=2
R37)    RAPID
        ONTCUT
        GO TO/U(K4)
        GOLTA/J,0,A
        CUT
        GOLTA/J,0,-(B+A),FOW
        FEDRAT/FO
        ONTCUT
        GO TO/V(K4)
        GOLTA/J,0,-B
        CUT
        GOLTA/J,0,(B+A),FUP
        K4=K4+1
        IF (K4-K3) R37,R37,R38
R38)    TERMAC
        PRINT/J
```

1.1.2.3. RGH2 (zigzag roughing macro)

```
RGH2=MACRO/

$$      MACRO RGH2 IS A ROUGHING MACRO ( IT USES THE ADDITIONAL MACROS  7 5
$$        MAC8 AND MAC9 )                                               7 6
$$        THE CUTTER ROUGHS OUT THE POCKET WHILE PERFORMING A ZIGZAG-   7 7
$$        MOTION FROM THE LEFT SIDE TO THE RIGHT SIDE OF THE POCKET     7 8
$$        ( AND VICA VERSA )                                            7 9

FPRINT  ***************************************************************
PPRINT
FPRINT  END OF THE SEMIROUGHING CUT
PPRINT
FPRINT  ***************************************************************

$$      STARTUP MACRO :                                                 7 10

        K3=0                                                            7 20
7A)     K3=K3+2*R                                                       7 30
        IF (K3-(Y1(K2)-Y1(1))) 7A,7B,7B                                 7 40

$$      K3 IS THE NUMBER OF PASSES                                      7 50

7B)     K3=K3/(2*R)                                                     7 60
        YA=Y1(1)                                                        7 160
        KR=K1                                                           7 170
        KL=2

$$      J=1         : GO TO MACRO 9                                     7 190
$$      J=-1        : GO TO MACRO 8                                     7 200

        J=1                                                             7 210
        I=0                                                             7 220
7H)     I=I+1                                                           7 230
        IF(I-K3) 7D,7D,7E                                               7 250
7D)     IF(J) 7F,7F,7G                                                  7 260
7G)     CALL/MAC9                                                       7 270
        J=-1                                                            7 280
        JUMPTO/7H                                                       7 290
7F)     CALL/MAC8                                                       7 300
        J=1                                                             7 310
        JUMPTO/7H                                                       7 320
7E)     GODLTA/J,0,(B+A),FUP                                            7 340
        TERMAC
        PRINT/0
```

```
MAC8=MACRO/                                                          8 10
$$      IN MACRO 8 THE TOOL CUTS ITS WAY TO THE LEFT HAND SIDE OF THE
$$      POCKET AND CLIMBS UP A DISTANCE OF TWICE ITS CUTTERRADIUS IN
$$      THE POSITIVE Y DIRECTION

8Y)     IF (YA-Y1(KL)) 8A,8B,8C                                      8 30
$$      ARRIVING BELOW Y1(KL)                                        8 20
8A)     IF (KS(KL-1)) 35,85,86                                       8 40
35)     INDIRV/-1,0,0                                                8 50
        GO/ON,CA(KL-1),(PLANE/0,0,1,-8),ON,(LINE/PARLEL,LX,YLARGE,YA) 8 60
        JUMPTO/8G                                                    8 70
86)     ONTCUT                                                       8 80
        GO/ON,(LINE/(POINT/CENTER,CA(KL-1)),PARLEL,LY),(PLANE/0,0,1, 8 90
          -8),ON,(LINE/PARLEL,LX,YLARGE,YA)                          8 91
        INDIRV/-1,0,0                                                8 100
        GO/ON,CA(KL-1),(PLANE/0,0,1,-8),ON,(LINE/PARLEL,LX,YLARGE,YA) 8 110
        CUT                                                          8 120
8G)     YA=YA+2*R                                                    8 130
        TT=-1                                                        8 140
        JUMPTO/879                                                   8 160
$$      ARRIVING AT Y1(KL)                                           8 170
88)     GO TO/(POINT/X1(KL),Y1(KL),-8)                               8 180
        YA=YA+2*R                                                    8 190
        ONTCUT
820)    CALL/MAC82                                                   8 200
        IF (W(KL)) 840,841,842                                       8 210
840)    TLON,GORGT/CA(KL-1),ON,CA(KL)                                8 220
        CUT                                                          8 230
        JUMPTO/801                                                   8 240
841)    TLON,GORGT/CA(KL-1),TANTO,CA(KL)                             8 250
        CUT                                                          8 260
        JUMPTO/801                                                   8 270
842)    TLON,GORGT/CA(KL-1),TANTO,(CIRCLE/CENTER,P(KL),RADIUS,R)     8 280
        CUT                                                          8 284
        IF (YA-Y2(KL)) 844,844,843                                   8 288
843)    TLON,GOFWD/(CIRCLE/CENTER,P(KL),RADIUS,R),TANTO,CA(KL)       8 292
        JUMPTO/801                                                   8 296
844)    TLON,GOFWD/(CIRCLE/CENTER,P(KL),RADIUS,R),ON,(LINE/PARLEL,LX,S 8 300
          YLARGE,YA)                                                 8 301
        JUMPTO/8H                                                    8 305
800)    TT=W(KL-1)                                                   8 310
879)    IF (W(KL))   870,871,872                                     8 320
870)    PP=0                                                         8 330
        CALL/MAC83,CO=ON                                             8 340
        IF (PP) 801,801,8H                                           8 360
871)    PP=0                                                         8 370
        CALL/MAC83,CO=TANTO                                          8 380
        IF (PP) 801,801,8H                                           8 400
872)    PP=0                                                         8 410
        CALL/MAC84                                                   8 420
        IF (PP) 801,801,8H                                           8 435
801)    KL=KL+1                                                      8 440
        IF(KL-K2) 900,800,8H                                         8 450
$$      ARRIVING ABOVE Y1(KL)                                        8 460
8C)     IF (YA-Y2(KL)) 850,851,852                                   8 470
$$      ARRIVING BELOW Y2(KL).                                       8 480
850)    INDIRV/-1,0,0                                                8 488
        GO/ON,(CIRCLE/CENTER,P(KL),RADIUS,R),(PLANE/0,0,1,-8),ON,(LINE 8 490
          /PARLEL,LX,YLARGE,YA)                                      8 491
        YA=YA+2*R                                                    8 500
        IF (YA-Y2(KL)) 846,846,845                                   8 503
845)    TLON,GORGT/(CIRCLE/CENTER,P(KL),RADIUS,R),TANTO,CA(KL)       8 506
        JUMPTO/801                                                   8 509
846)    TLON,GORGT/(CIRCLE/CENTER,P(KL),RADIUS,R),ON,(LINE/PARLEL,LX,S 8 512
          YLARGE,YA)                                                 8 513
                                                                     8 516
```

```
$$      ARRIVING AT Y2(KL)                                                  8  520

851)    INDIRV/-1,0,0                                                       8  530
        GO TO/(POINT/X2(KL),Y2(KL),-8)                                      8  540
        YA=YA+2*R                                                           8  550
        DNTCUT                                                              8  555
        CALL/MAC82                                                          8  560
        TLON,GORGT/CA(KL-1),TANTO,(CIRCLE/CENTER,P(KL),RADIUS,R)            8  570
        TLON,GOFWD/(CIRCLE/CENTER,P(KL),RADIUS,R),TANTO,CA(KL)              8  580
        CUT                                                                 8  590
        JUMPTO/801                                                          8  600

$$      ARRIVING ABOVE Y2(KL)                                               8  610

852)    KL=KL+1                                                             8  620
        JUMPTO/8Y                                                           8  630
8H)     TERMAC                                                              8  550
        PRINT/0


        MAC83=MACRO/QQ                                                     83  10

$$      WHILE CLIMBING UP ON THE LEFT  HAND SIDE( OVER A DISTANCE OF
$$      2*R ), THE CUTTER MIGHT ENCOUNTER SEVERAL DRIVESURFACES.
$$      FOR EVERY SUCH DRIVESURFACE ( WHICH WILL NOT GIVE YOU A CONCA-
$$      VITY ), MACRO MAC83 IS CALLED UP BY MAC8

        IF (TT) 873,874,874                                                83  20
873)    IF (YA-Y1(KL))   810,810,811                                       83  30
810)    TLON,GORGT/CA(KL-1),ON,(LINE/PARLEL,LX,YLARGE,YA)                  83  40
        PP=1                                                               83  50
        JUMPTO/814                                                         83  60
811)    TLON,GORGT/CA(KL-1),QQ,CA(KL)                                      83  70
        PP=-1                                                              83  80
        JUMPTO/814                                                         83  90
874)    IF (YA-Y1(KL))   812,812,813                                       83  100
812)    TLON,GOFWD/CA(KL-1),ON,(LINE/PARLEL,LX,YLARGE,YA)                  83  110
        PP=1                                                               83  120
        JUMPTO/814                                                         83  130
813)    TLON,GOFWD/CA(KL-1),QC,CA(KL)                                      83  140
        PP=-1                                                              83  150
814)    TERMAC                                                             83  160
        PRINT/0


        MAC82=MACRO/                                                       82  10

$$      IN THIS MACRO THE CUTTER PRETENDS TO GO BACK TO THE PREVIOUS        82  20
$$        INTERSECTION POINT P(KL-1) AND TO WORK ITS WAY BACK UP TO         82  21
$$        LEVEL YA                                                         82  22

        KK=KL-1                                                            82  30
        GO TO/(POINT/X2(KK),Y2(KK),0)                                      82  40
        IF (A2(KK)) 824,825,826                                            82  50
824)    AN=A2(KK)+90                                                       82  60
        JUMPTO/830                                                         82  70
826)    AN=A2(KK)-90                                                       82  80
        JUMPTO/830                                                         82  90
825)    IF (X1(KL)-X2(KK)) 828,829,827                                     82  100
829)    IF (Y1(KL)-Y2(KK)) 827,828,828                                     82  110
828)    AN=-90                                                             82  120
        JUMPTO/830                                                         82  130
827)    AN=+90                                                             82  140
830)    GODLTA/(.1*COSF(AN)),(.1*SINF(AN)),0                               82  150
        AA1=-1*COSF(AN)                                                    82  154
        AA2=-1*SINF(AN)                                                    82  158
        INDIRV/(VECTOR/AA1,AA2,0)                                          82  160
        IF (A2(KK)) 831,832,831                                            82  170
832)    IF (KS(KL-1)) 831,833,831                                          82  180
833)    GO/ON,CA(KL-1),(PLANE/0,0,1,-8),ON,(LINE/PARLEL,LY,XLARGE,         82  190
          X2(KK))                                                          82  191
        JUMPTO/834                                                         82  200
831)    GO/ON,CA(KL-1),(PLANE/0,0,1,-8),ON,(LINE/PARLEL,LX,YLARGE,         82  210
          Y2(KK))                                                          82  211
834)    TERMAC                                                             82  220
```

```
$$      IN MACRO 9 THE TCOL CUTS ITS WAY TO THE RIGHT HAND SIDE OF THE
$$      POCKET AND CLIMBS UP A DISTANCE OF TWICE ITS CUTTERRADIUS IN
$$      THE POSITIVE Y DIRECTION

9Y)     IF (YA-Y2(KR)) 9A,9B,9C                                          9 20
$$      ARRIVING BELOW Y2(KR)                                            9 30
9A)     IF (I-1) 91,91,92                                                9 40
$$      FIRST PASS                                                       9 50
91)     IF (W(1)) 93,94,94                                               9 60
93)     CALL/ADD1,II=-.1,JJ=0,KK=1,LL=0                                  9 70
        JUMPTO/9G                                                        9 80
94)     CALL/ADD1,II=0   ,JJ=.1,KK=0,LL=-1                               9 84
        JUMPTO/9G                                                        9 88
$$      FUTHER PASSES                                                    9 90
92)     IF(KS(KR)) 95,95,96                                              9 100
95)     INDIRV/1,0,0                                                     9 110
        GO/ON,CA(KR),(PLANE/0,0,1,-B),ON,(LINE/PARLEL,LX,YLARGE,YA)      9 120
        JUMPTO/9G                                                        9 130
96)     DNTCUT                                                           9 140
        GO/ON,(LINE/(POINT/CENTER,CA(KR)),PARLEL,LY),(PLANE/0,0,1,-B)$   9 150
          ,ON,(LINE/PARLEL,LX,YLARGE,YA)                                 9 151
        INDIRV/1,0,0                                                     9 160
        GO/ON,CA(KR),(PLANE/0,0,1,-B),ON,(LINE/PARLEL,LX,YLARGE,YA)      9 170
        CUT                                                              9 180
9G)     YA=YA+2*R                                                        9 190
        TT=-1                                                            9 210
        JUMPTO/979                                                       9 220
$$      ARRIVING AT Y2(KR)                                               9 230
98)     GO TO/(POINT/X2(KR),Y2(KR),-B)                                   9 240
        YA=YA+2*R                                                        9 250
        DNTCUT                                                           9 255
920)    CALL/MAC92                                                       9 260
        IF (W(KR)) 940,941,942                                           9 270
940)    TLON,GOLFT/CA(KR),ON,CA(KR-1)                                    9 280
        CUT                                                              9 290
        JUMPTO/901                                                       9 300
941)    TLON,GOLFT/CA(KR),TANTO,CA(KR-1)                                 9 310
        CUT                                                              9 320
        JUMPTO/901                                                       9 330
942)    TLON,GOLFT/CA(KR),TANTO,(CIRCLE/CENTER,P(KR),RADIUS,R)           9 340
        CUT                                                              9 350
        IF (YA-Y1(KR)) 944,944,943                                       9 353
943)    TLON,GOFWD/(CIRCLE/CENTER,P(KR),RADIUS,R),TANTO,CA(KR-1)         9 356
        JUMPTO/901                                                       9 359
944)    TLON,GOFWD/(CIRCLE/CENTER,P(KR),RADIUS,R),ON,(LINE/PARLEL,LX,$   9 362
          YLARGE,YA)                                                     9 363
        JUMPTO/9H                                                        9 366
900)    IF (KR-K1) 961,960,960                                           9 368
960)    KK=1                                                             9 371
        JUMPTO/962                                                       9 372
961)    KK=KR+1                                                          9 374
962)    TT=W(KK)                                                         9 376
979)    IF (W(KR))   970,971,972                                         9 380
970)    PP=0                                                             9 390
        CALL/MAC93,OO=ON                                                 9 395
        IF (PP) 901,911,9H                                               9 410
971)    PP=0                                                             9 420
        CALL/MAC93,OO=TANTO                                              9 425
        IF (PP) 901,911,9H                                               9 440
972)    PP=0                                                             9 450
        CALL/MAC94                                                       9 460
        IF (PP) 901,901,9H                                               9 480
901)    KR=KR-1                                                          9 490
        IF (KR-K2) 9H,900,900
$$      ARRIVING ABOVE Y2(KR)                                            9 510
9C)     IF (YA-Y1(KR)) 950,951,952                                       9 520
$$      ARRIVING BELOW Y1(KR)                                            9 530
```

```
                                                                        9 550
                                                                        9 551
                                                                        9 560
        YA=YA+2*R                                                       9 563
        IF (YA-Y1(KR)) 946,946,945                                      9 566
945)    TLON,GOLFT/(CIRCLE/CENTER,P(KR),RADIUS,R),TANTO,CA(KR-1)        9 569
        JUMPTO/951
946)    TLON,GOLFT/(CIRCLE/CENTER,P(KR),RADIUS,R),ON,(LINE/PARLEL,LX,   9 572
          YLARGE,YA)                                                    9 573
        JUMPTO/9H                                                       9 575

$$      ARRIVING AT Y1(KR)                                              9 580

951)    INDIRV/1,0,0                                                    9 590
        GO TO/(POINT/X1(KR),Y1(KR),-B)                                  9 600
        YA=YA+2*B                                                       9 610
        ONTCUT                                                          9 615
        CALL/MAC92                                                      9 620
        TLON,GOLFT/CA(KR),TANTO,(CIRCLE/CENTER,P(KR),RADIUS,R)          9 630
        TLON,GOFWD/(CIRCLE/CENTER,P(KR),RADIUS,R),TANTO,CA(KR-1)        9 640
        CUT                                                             9 650
        JUMPTO/951                                                      9 660

$$      ARRIVING ABOVE Y1(KR)                                           9 670

952)    KR=KR-1                                                         9 680
        JUMPTO/9Y                                                       9 690
9H)     TERMAC                                                          9 710
        PRINT/0

        MAC92=MACRO/                                                    92 10

$$      IN THIS MACRO, THE CUTTER PRETENDS TO GO BACK TO THE PREVIOUS   92 20
$$        INTERSECTION POINT  (KR+1) AND TO WORK ITS WAY UP TO THE      92 21
$$        LEVEL YA                                                      92 22

        IF (KR-K1) 921,922,922                                          92 30
921)    KK=KR+1                                                         92 40
        JUMPTO/923                                                      92 50
922)    KK=1                                                            92 60
923)    GO TO/(POINT/X1(KK),Y1(KK),0)                                   92 70
        IF (A1(KK)) 924,925,924                                         92 80
924)    AN=A1(KK)+90                                                    92 90
        JUMPTO/930                                                      92 120
925)    IF (X2(KR)-X1(KK)) 928,929,927                                  92 130
929)    IF (Y2(KR)-Y1(KK)) 927,927,928                                  92 140
927)    AN=90                                                           92 150
        JUMPTO/930                                                      92 160
928)    AN=-90                                                          92 170
930)    GODLTA/(.1*COSF(AN)),(.1*SINF(AN)),0                            92 180
        AA1=-1*COSF(AN)                                                 92 184
        AA2=-1*SINF(AN)                                                 92 188
        INDIRV/(VECTOR/AA1,AA2,0)                                       92 190
        IF (A1(KK)) 931,932,931                                         92 200
932)    IF (KS(KR)) 931,933,931                                         92 210
933)    GO/CN,CA(KR),(PLANE/0,0,1,-B),ON,(LINE/PARLEL,LY,XLARGE,X1(KK)) 92 220
        JUMPTO/934                                                      92 230
931)    GO/CN,CA(KR),(PLANE/0,0,1,-B),ON,(LINE/PARLEL,LX,YLARGE,Y1(KK)) 92 240
934)    TERMAC                                                          92 250

        MAC93=MACRO/00                                                  93 10

$$      WHILE CLIMBING UP ON THE RIGHT HAND SIDE( OVER A DISTANCE OF
$$      2*R ), THE CUTTER MIGHT ENCOUNTER SEVERAL DRIVESURFACES.
$$      FOR EVERY SUCH DRIVESURFACE ( WHICH WILL NOT GIVE YOU A CONCA-
$$      VITY ), MACRO MAC93 IS CALLED UP BY MAC9

        IF (TT) 973,974,974                                             93 20
973)    IF (YA-Y2(KR))   910,910,911                                    93 30
910)    TLON,GOLFT/CA(KR  ),ON,(LINE/PARLEL,LX,YLARGE,YA)               93 40
        PP=1                                                            93 50
        JUMPTO/914                                                      93 60
911)    TLON,GOLFT/CA(KR  ),OO,CA(KR-1)                                 93 70
        PP=-1                                                           93 80
        JUMPTO/914                                                      93 90
974)    IF (YA-Y2(KR  )) 912,912,913                                    93 100
912)    TLON,GOFWD/CA(KR  ),ON,(LINE/PARLEL,LX,YLARGE,YA)               93 110
        PP=1                                                            93 120
        JUMPTO/914                                                      93 130
913)    TLON,GOFWD/CA(KR  ),OO,CA(KR-1)                                 93 140
        PP=-1                                                           93 150
914)    TERMAC                                                          93 160
```

```
                                                                        84 10
$$      WHILE CLIMBING UP THE LEFT  HAND SIDE, THE CUTTER USES THE
$$      SURFACES OF THE POCKET AS DRIVESURFACES
$$      FOR EVERY DRIVESURFACE WHICH WILL GIVE THE CUTTER DIFFICULTIES
$$      BECAUSE OF CONCAVITY, MACRO MAC84 IS CALLED UP BY MAC8 INSTEAD
$$      OF MAC83.

        IF (TT) 877,878,878                                             84 20
877)    IF (YA-Y1(KL)) 815,815,816                                      84 30
815)    TLON,GORGT/CA(KL-1),ON,(LINE/PARLEL,LX,YLARGE,YA)               84 40
        PP=1                                                            84 50
        JUMPTO/803                                                      84 60
816)    TLON,GORGT/CA(KL-1),TANTO,(CIRCLE/CENTER,P(KL),RADIUS,R)        84 70
        JUMPTO/819                                                      84 80
878)    IF (YA-Y1(KL)) 817,817,818                                      84 90
817)    TLON,GOFWD/CA(KL-1),ON,(LINE/PARLEL,LX,YLARGE,YA)               84 100
        PP=1                                                            84 110
        JUMPTO/803                                                      84 120
818)    TLON,GOFWD/CA(KL-1),TANTO,(CIRCLE/CENTER,P(KL),RADIUS,R)        84 130
819)    IF (YA-Y2(KL)) 804,804,805                                      84 140
804)    TLON,GOFWD/(CIRCLE/CENTER,P(KL),RADIUS,R),ON,(LINE/PARLEL,   $  84 150
          LX,YLARGE,YA)                                                 84 151
        PP=1                                                            84 152
        JUMPTO/803                                                      84 170
805)    TLON,GOFWD/(CIRCLE/CENTER,P(KL),RADIUS,R),TANTO,CA(KL)          84 180
        PP=-1                                                           84 190
803)    TERMAC                                                          84 200
        PRINT/0


        MAC94=MACRO/                                                    94 10

$$      WHILE CLIMBING UP THE RIGHT HAND SIDE, THE CUTTER USES THE
$$      SURFACES OF THE POCKET AS DRIVESURFACES
$$      FOR EVERY DRIVESURFACE WHICH WILL GIVE THE CUTTER DIFFICULTIES
$$      BECAUSE OF CONCAVITY, MACRO MAC94 IS CALLED UP BY MAC9 INSTEAD
$$      OF MAC93.

        IF (TT) 977,978,978                                             94 20
977)    IF (YA-Y2(KR)) 915,915,916                                      94 30
915)    TLON,GOLFT/CA(KR),ON,(LINE/PARLEL,LX,YLARGE,YA)                 94 40
        PP=1                                                            94 50
        JUMPTO/903                                                      94 60
916)    TLON,GOLFT/CA(KR),TANTO,(CIRCLE/CENTER,P(KR),RADIUS,R)          94 70
        JUMPTO/919                                                      94 80
978)    IF (YA-Y2(KR)) 917,917,918                                      94 90
917)    TLON,GOFWD/CA(KR),0.1,(LINE/PARLEL,LX,YLARGE,YA)                94 100
        PP=1                                                            94 110
        JUMPTO/903                                                      94 120
918)    TLON,GOFWD/CA(KR),TANTO,(CIRCLE/CENTER,P(KR),RADIUS,R)          94 130
919)    IF (YA-Y1(KR)) 904,904,905                                      94 140
904)    TLON,GOFWD/(CIRCLE/CENTER,P(KR),RADIUS,R),ON,(LINE/PARLEL,    $  94 150
          LX,YLARGE,YA)                                                 94 151
        PP=1                                                            94 160
        JUMPTO/903                                                      94 170
905)    TLON,GOFWD/(CIRCLE/CENTER,P(KR),RADIUS,R),TANTO,CA(KR-1)        94 180
        PP=-1                                                           94 190
903)    TERMAC                                                          94 200
        PRINT/0


        ADD1=MACRO/II,JJ,KK,LL                                          A1 10

$$      MACRO IN ORDER TO OBTAIN THE FOLLOWING STARTUP OF THE TOOL :    A1 20
$$        TLON,GOLFT ALONG ROUGHING SURFACE CS(K1)                      A1 21

        CNTOUT                                                          A1 30
        GODLTA/II,JJ,0                                                  A1 40
        INDIRV/KK,LL,0                                                  A1 50
        GO/ON,CA(KR),(PLANE/0,0,1,-R),ON,(LINE/PARLEL,LX,YLARGE,YA)     A1 60
        CUT                                                             A1 70
        TERMAC                                                          A1 80
        PRINT/0
```

1.1.2.4. SPIRAL

```
        SPIR1=MACRO/JJ                                                  S1 1
$$      ADDITIONAL MACRO TO SPIR2                                       S1 5
$$           CALCULATE THE FOUR INNER ANGLES
$$      0≤H(JJ)<180
        IF(A2(JJ))SA1,SA2,SA2                                           S1 10
SA1)    H(JJ)=A2(JJ)+180                                                S1 20
        JUMPTO/SA5                                                      S1 30
SA2)    H(JJ)=A2(JJ)                                                    S1 40
SA5)    IF(A1(JJ+1))SA3,SA3,SA4                                         S1 50
SA3)    H(JJ+1)=-A1(JJ+1)                                               S1 60
        JUMPTO/SA6                                                      S1 70
SA4)    H(JJ+1)=180-A1(JJ+1)                                            S1 80
SA6)    TERMAC                                                          S1 90
        PRINT/0

        SPIR2=MACRO/J                                                   S2 1
$$      DETERMINING THE CB(J) CHECKSURFACES                             S2 5
        IF(KS(J))SB1,SB2,SB3                                            S2 10
$$      CONCAVE CIRCLE REMAINS THE SAME                                 S2 15
SB1)    CALL/SPIR1,JJ=J                                                 S2 20
        CB(J)=CIRCLE/CS(J),CANON,,,,,,,                                 S2 30
        JUMPTO/SB4                                                      S2 40
$$      LINE REMAINS THE SAME                                           S2 45
SB2)    CALL/SPIR1,JJ=J                                                 S2 50
        CB(J)=LINE/CS(J),CANON,,,,                                      S2 60
        JUMPTO/SB4                                                      S2 70
$$      CONVEX CIRCLE BECOMES A LINE                                    S2 75
SB3)    CB(J)=LINE/P(J),P(J+1)                                          S2 80
        OBTAIN,LINE/CB(J),AX,BX,,
        IF (BX) SB7,SB8,SB7                                             S  100
SB8)    H(J)=90                                                         S  102
        JUMPTO/SB9                                                      S  104
SB7)    H(J)=ATANF(-AX/BX)                                              S  106
SB9)    H(J+1)=180-H(J)                                                 S  110
        IF (J-2) SB5,SB5,SB6                                            S2 120
SB5)    CALL/SPIR3,JJ=J,AA=XSMALL                                       S2 130
        JUMPTO/SB4                                                      S2 140
SB6)    CALL/SPIR3,JJ=J,AA=XLARGE                                       S2 150
SB4)    TERMAC                                                          S2 160
        PRINT/0

        SPIR3=MACRO/JJ,AA                                               S3 1
$$      CALCULATING THE MAXIMUM DISTANCE ( IN WIDTH ) YET TO BE CUT     S3 5
        L((JJ+1)/2)=LINE/(POINT/AA,INTOF,(LINE/(POINT/CENTER,CS(JJ))$   S3 10
          ,PERPTO,CB(JJ)),CS(JJ)),PARLEL,CB(JJ)                         S3 11
        OBTAIN,LINE/CB(JJ),,,,D1                                        S3 20
        OBTAIN,LINE/L((JJ+1)/2),,,,D2                                   S3 30
        DIS((JJ+1)/2)=ABSF(D1-D2)                                       S3 40
        TERMAC                                                          S3 50
        PRINT/0
```

```
        SPIR4=MACRO/                                                    S4 1
$$      FIRST, CALCULATE THE SMALLEST ANGLE, I.E. AX AND CALCULATE DEL  S4 10
        AX=H(1)
        J=2                                                             S4 30
SD3)    IF(H(J)-AX) SD1,SD2;SD2
SD1)    AX=H(J)
SD2)    J=J+1                                                           S4 60
        IF (J-4) SD3,SD3,SD4                                            S4 70
SD4)    DEL=R*(1-SINF(AX/2))/2
        P1=R-DEL                                                        S4 90
$$      SECONDLY, CALCULATE THE NUMBER OF PASSES, I.E. K3               S4 100
        K3=2*DEL                                                        S4 110
SD5)    K3=K3+2*R1                                                      S4 120
        IF (K3-(Y(2)-Y(1))) SD5,SD6,SD7                                 S4 130
SD6)    K3=(K3-2*DEL)/(2*R1)                                            S4 140
        JUMPTO/SD8                                                      S4 150
SD7)    K3=(K3-2*R1-2*DEL)/(2*R1)                                       S4 160
$$      THIPOLY, FIND OUT WHETHER OR NOT IT IS AN EVEN OR UNEVEN        S4 170
$$        NUMBER OF PASSES                                              S4 171
$$      ALSO       K4=K3/2                                            S4 172
$$                 K5=INT(K3/2)                                       S4 173
SD8)    K4=K3/2                                                         S4 180
        K5=0                                                            S4 190
SD9)    K5=K5+1                                                         S4 200
        IF (K5-K4) SD9,SD10,SD11                                        S4 210
SD10)   EX=1                                $$  EVEN
        JUMPTO/SD12                                                     S4 230
SD11)   EX=-1                               $$  UNEVEN
        K5=K5-1
SD12)   CALL/SPIR5,JJ=1,AA=XLARGE                                       S4 250
        CALL/SPIR5,JJ=3,AA=XSMALL                                       S4 260
        CB(4)=LINE/CS(4),CANON,,,,                                      S4 270
/       CALL/SPIR5,JJ=4,AA=YLARGE                                       S4 275
        TERMAC                                                          S4 280
        PRINT/0                                                         S4 290

        SPIR5=MACRO/JJ,AA                                               S5 1
$$      CALCULATING THE NEW CB(JJ)                                      S5 5
        IF(KS(JJ))SE1,SE2,SE2                                           S5 10
SE1)    OBTAIN,CIRCLE/CB(JJ),,,,,,,AX                                   S5 20
        CB(JJ)=CIRCLE/CB(JJ),CANON,,,,,,,(AX+DEL)                       S5 30
        JUMPTO/SE3                                                      S5 40
SE2)    LA(JJ)=LINE/PARLEL,CB(JJ),AA,DEL                                S5 45
        CB(JJ)=LINE/LA(JJ),CANON,,,,                                    S5 55
SE3)    TERMAC                                                          S5 60
        PRINT/0

        SPIR6=MACRO/JJ                                                  S6 1
$$      SUBROUTINE USED TO COUNT THE NUMBER OF PASSES IN THE LEFT OR    S6 10
$$      RIGHT SIDE AREA.                                                S6 11
        K7=0                                                            S6 20
SF3)    K7=K7+1                                                         S6 30
        IF (DIS(JJ)-K7*2*R) SF2,SF2,SF3                                 S6 40
SF2)    TERMAC                                                          S6 50
        PRINT/0
```

```
          SPIR7=MACRO/                                                    S7 1
$$        SPIR7 IS USED TO ROUGH OUT THE UNCUT AREA IN BETWEEN CS(1) AND  S7 4
$$        CB(1) IN CASE THE SURFACE CS(1) IS CONVEX                       S7 5
          GODLTA/0,0,(B+A),FUP                                            S7 10
          RAPID                                                           S7 20
          GO TO/(POINT/YSMALL,INTOF,(LINE/PARLEL,CB(1),XSMALL,((2*K8-1)   SS7 30
          *R+DEL)),CA(1))                                                 S7 31
          GODLTA/0,0,-(B+A),FDW                                           S7 40
          PSIS/(PLANE/0,0,1,-B)                                           S7 90
          FEDRAT/FD                                                       S7 100
          GO TO/(POINT/YLARGE,INTOF,(LINE/PARLEL,CB(1),XSMALL,((2*K8-1)   SS7 110
          *R+DEL)),CA(1))                                                 S7 111
          TERMAC                                                          S7 120
          PRINT/0

          SPIR8=MACRO/                                                    S8 1
$$        SPIR8 IS USED TO ROUGH OUT THE UNCUT AREA IN BETWEEN CS(3) AND  S8 4
$$        CB(3) IN CASE THE SURFACE CS(3) IS CONVEX                       S8 5
          GODLTA/0,0,(B+A),FUP                                            S8 10
          RAPID                                                           S8 20
          GO TO/(POINT/YLARGE,INTOF,(LINE/PARLEL,CB(3),XLARGE,((2*K8-1)   SS8 30
          *R+DEL)),CA(3))                                                 S8 31
          GODLTA/0,0,-(B+A),FDW                                           S8 40
          PSIS/(PLANE/0,0,1,-B)                                           S8 90
          FEDRAT/FD                                                       S8 100
          GO TO/(POINT/YSMALL,INTOF,(LINE/PARLEL,CB(3),XLARGE,((2*K8-1)   SS8 110
          *R+DEL)),CA(3))                                                 S8 111
          TERMAC                                                          S8 120
          PRINT/0

          SPIR9=MACRO/
$$        MACRO SPIR9 IS TO POSITION THE CUTTER IN THE UNCUT AREA ON THE  SP A2
$$        LEFT  HAND SIDE (IN CASE OF A CONVEX SURFACE CS(1)) AND TO TAKE SP A3
$$        THE FIRST CUT IN THAT AREA                                      SP A4
$$        (THIS MACRO WAS NEEDED BECAUSE OF ERROR MESSAGE 12 , I.E.       SP A5
$$        MACRO SPIRAL TOO LARGE)                                         SP A6
PPRINT    **************************************************************
PPRINT    ROUGH OUT THE MATERIAL SITUATED ON THE LEFT HAND SIDE OF THE
PPRINT    POCKET ( WITHIN THE CONVEX CIRCLE SECTOR )
PPRINT    **************************************************************
          RAPID                                                           SP 200
          GO TO/(POINT/X1(1),Y1(1),A)                                     SP 810
          GODLTA/0,0,-(B+A),FDW                                           SP 820
          ONTCUT                                                          SP 830
          GODLTA/.01,0,0                                                  SP 845
          INDIRV/-1,0,0                                                   SP 850
          GO/ON,CA(1),(PLANE/0,0,1,-B),ON,(LINE/PARLEL,LX,YLARGE,Y1(1))   SP 855
          CUT                                                             SP 860
          FEDRAT/FD                                                       SP 865
          IF (W(1)) S42,S43,S43                                           SP 870
S42)      TLON,GORGT/CA(1),ON,(LINE/PARLEL,CB(1),XSMALL,(R+DEL)),         SP 875
          JUMPTO/S44                                                      SP 880
S43)      TLON,GOFWD/CA(1),ON,(LINE/PARLEL,CB(1),XSMALL,(R+DEL))          SP 885
S44)      TLON,GORGT/(LINE/PARLEL,CB(1),XSMALL,(R+DEL)),TO,CA(1)          SP 890
          K8=1                                                            SP 910
S28)      K8=K8+1                                                         SP 920
          IF (K7-K8) S26,S26,S27                                          SP 930
S26)      GODLTA/0,0,(B+A),FUP                                            SP 940
          JUMPTO/S40                                                      SP 950
S27)      CALL/SPIR7                                                      SP 960
          JUMPTO/S28                                                      SP 970
S40)      TERMAC
          PRINT/0
```

```
          SPIR10=MACRO/                                                          B 20
$$        MACRO SPIR10 IS TO POSITION THE CUTTER IN THE UNCUT AREA ON THESP  B2
$$        RIGHT HAND SIDE (IN CASE OF A CONVEX SURFACE CS(3)) AND TO TAKESP  B3
$$        THE FIRST CUT IN THAT AREA                                       SP  B4
$$        (THIS MACRO WAS NEEDED BECAUSE OF ERROR MESSAGE 12 , I.E.        SP  B5
$$        MACRO SPIRAL TOO LARGE)                                          SP  B6
PPRINT    **********************************************************************
PPRINT    ROUGH OUT THE MATERIAL SITUATED ON THE RIGHT HAND SIDE OF THE
PPRINT    POCKET ( WITHIN THE CONVEX CIRCLE SECTOR)
PPRINT    **********************************************************************
          RAPID                                                                SP 1010
          GO TO/(POINT/X1(3),Y1(3),A)                                          SP 1020
          GODLTA/J,0,-(B+A),FDW                                                SP 1030
          ONTCUT                                                               SP 1040
          GODLTA/-.01,0,0                                                      SP 1050
          INDIRV/1,0,0                                                         SP 1060
          GO/ON,CA(3),(PLANE/0,0,1,-B),ON,(LINE/PARLEL,LX,YLARGE,Y1(3))        SP 1065
          CUT                                                                  SP 1070
          FEDRAT/FD                                                            SP 1075
          IF (W(3)) S52,S53,S53                                                SP 1080
S52)      TLON,GORGT/CA(3),ON,(LINE/PARLEL,CB(3),XLARGE,(R+DEL))               SP 1085
          JUMPTO/S54                                                           SP 1090
S53)      TLON,GOFWD/CA(3),ON,(LINE/PARLEL,CB(3),XLARGE,(R+DEL))               SP 1095
S54)      TLON,GORGT/(LINE/PARLEL,CB(3),XLARGE,(R+DEL)),TO,CA(3)               SP 1100
          K8=1                                                                 SP 1120
S34)      K8=K8+1                                                              SP 1130
          IF (K7-K8) S32,S32,S33                                               SP 1140
S32)      GODLTA/J,0,(B+A),FUP                                                 SP 1150
          JUMPTO/S41                                                           SP 1160
S33)      CALL/SPIR8                                                           SP 1170
          JUMPTO/S34                                                           SP 1180
S41)      TERMAC
          PRINT/0
          SPIR11=MACRO/                                                        S11 10
PPRINT    **********************************************************************
PPRINT    SEMIROUGHING CUT ON LEFT HAND SIDE OF THE POCKET
PPRINT    **********************************************************************
          RAPID                                                                S11 20
          GO TO/(POINT/X1(1),Y1(1),A)                                          S11 30
          GODLTA/0,0,-(B+A),FDW                                                S11 40
          ONTCUT                                                               S11 50
          GODLTA/.01,0,0                                                       S11 60
          INDIRV/-1,0,0                                                        S11 70
          GO/ON,CA(1),(PLANE/0,0,1,-B),ON,(LINE/PARLEL,LX,YLARGE,Y1(1))        S11 80
          CUT                                                                  S11 90
          FEDRAT/FD                                                            S11 100
          IF (W(1)) S62,S63,S63                                                S11 110
S62)      TLON,GORGT/CA(1),ON,(LINE/PARLEL,LX,YLARGE,Y1(2))                    S11 120
          JUMPTO/S64                                                           S11 130
S63)      TLON,GOFWD/CA(1),ON,(LINE/PARLEL,LX,YLARGE,Y1(3))                    S11 140
S64)      GODLTA/J,0,(B+A),FUP                                                 S11 150
          TERMAC                                                               S11 160
          PRINT/0
          SPIR12=MACRO/                                                        S12 10
PPRINT    **********************************************************************
PPRINT    SEMIROUGHING CUT ON RIGHT HAND SIDE OF THE POCKET
PPRINT    **********************************************************************
          ONTCUT                                                               S12 50
          GODLTA/-.01,0,0                                                      S12 60
          INDIRV/1,0,0                                                         S12 70
          GO/ON,CA(3),(PLANE/0,0,1,-B),ON,(LINE/PARLEL,LX,YLARGE,Y1(3))        S12 80
          CUT                                                                  S12 90
          FEDRAT/FD                                                            S12 100
          IF (W(3)) S72,S73,S73                                                S12 110
S72)      TLON,GORGT/CA(1),ON,(LINE/PARLEL,LX,YLARGE,Y1(0))                    S12 120
          JUMPTO/S74                                                           S12 130
S73)      TLON,GOFWD/CA(3),ON,(LINE/PARLEL,LX,YLARGE,Y1(4))                    S12 140
S74)      GODLTA/J,0,(B+A),FUP                                                 S12 150
          TERMAC                                                               S12 160
```

```
          SPIRAL=MACRO/                                                    SP  1

$$        PREPARATORY CALCULATIONS DONE IN VARIOUS MACROS                  SP  10

          CALL/SPIR2,J=1                                                   SP  20
          CALL/SPIR2,J=3                                                   SP  30
          CALL/SPIR4                                                       SP  40

$$        CALCULATE THE LINES ON WHICH THE CUTTER WILL GO                  SP  50

          IF(KS(1))S1,S2,S2                                                SP  60
S1)       OBTAIN,CIRCLE/CB(1),,,,,,,AX                                     SP  70
          J=1                                                              SP  80
S3)       CD(J)=CIRCLE/CB(1),CANON,,,,,,,(AX+(2*J-1)*R1)                   SP  90
          J=J+1                                                            SP 100
          IF (J-(K5+1)) S3,S3,S4                                           SP 110
S2)       J=1                                                              SP 120
S5)       CD(J)=LINE/PARLEL,CB(1),XLARGE,((2*J-1)*R1)                      SP 130
          J=J+1                                                            SP 140
          IF (J-(K5+1)) S5,S5,S4                                           SP 150
S4)       IF(KS(3))S6,S7,S7                                                SP 160
          J=1                                                              SP 170
S6)       OBTAIN,CIRCLE/CB(3),,,,,,,AX                                     SP 180
          J=1                                                              SP 190
S8)       CE(J)=CIRCLE/CB(3),CANON,,,,,,,(AX+(2*J-1)*R1)                   SP 200
          J=J+1                                                            SP 210
          IF (J-K5) S8,S8,S9                                               SP 220
S7)       J=1                                                              SP 230
S10)      CE(J)=LINE/PARLEL,CB(3),XSMALL,((2*J-1)*R1)                      SP 240
          J=J+1                                                            SP 250
          IF (J-K5) S10,S10,S9                                             SP 260
S9)       J=1                                                              SP 270
S11)      CF(J)=LINE/PARLEL,CB(4),YLARGE,((2*J-1)*R1)                      SP 280
          J=J+1                                                            SP 290
          IF (J-K3) S11,S11,S12                                            SP 300
S12)      SA=K5                                                            SP 310
          IF (EX) S133,S132,S132                                           SP 320
S133)     SA=SA+1                                                          SP 325
S132)     IF (KS(1)) S13,S14,S14                                           SP 327
S13)      PP=POINT/XLARGE,INTOF,CF(SA),CD(K5+1)                            SP 330
          JUMPTO/S15                                                       SP 340
S14)      PP=POINT/INTOF,CC(K5+1),CF(SA)                                   SP 350
S15)      PP=POINT/PP,CANON,,,A                                            SP 360

$$        ACTUAL MOTION STATEMENTS                                         SP 370

          RAPID                                                            SP 375
          GO TO/PP                                                         SP 395
          GODLTA/J,0,-(B+A),FDW                                            SP 400
          PSIS/(PLANE/0,0,1,-B)                                            SP 410
          IF(EX)S16,S16,S17                                                SP 450

PPRINT    **********************************************************
PPRINT    BEGINNING OF THE SPIRAL
PPRINT
PPRINT    **********************************************************

$$        UNEVEN NUMBER OF PASSES                                          SP 465

S16)      ONTCUT                                                           SP 470

          GODLTA/0,10,0                                                    SP 473
          INDIRV/0,-1,0                                                    SP 475
          GO/ON,CF(K5+1),(PLANE/0,0,1,-B),ON,CD(K5+1)                      SP 480
          CUT                                                              SP 490
          K6=0                                                             SP 500
          FEDRAT/FD                                                        SP 510
S19)      TLON,GOLFT/CF(K5+1-K6),ON,CE(K5-K6)                              SP 520
          TLON,GOLFT/CE(K5-K6),ON,CF(K5+2+K6)                              SP 530
          TLON,GOLFT/CF(K5+2+K6),ON,CC(K5-K6)                              SP 540
          TLON,GOLFT/CC(K5-K6),ON,CF(K5-K6)                                SP 550
```

```
PPRINT  ***********************************************************
PPRINT
PPRINT  A SPIRAL PASS HAS JUST BEEN COMPLETED
PPRINT
PPRINT  ***********************************************************

        K6=K6+1                                                          SP 550
        IF (K5-K6) S18,S19,S19                                           SP 570
S18)    TLON,GOLFT/CF(1),ON,CE(1)                                        SP 580
        GODLTA/0,0,(B+A),FUP                                             SP 590
        JUMPTO/S22                                                       SP 600

$$      EVEN NUMBER OF PASSES                                            SP 605

S19)    ONTCUT                                                           SP 610
        GODLTA/0,10,0                                                    SP 613
        INDIRV/0,-1,0                                                    SP 615
        GO/ON,CF(K5),(PLANE/0,0,1,-B),ON,CO(K5+1)                        SP 620
        CUT                                                              SP 630
        FEDRAT/FD                                                        SP 640
        TLON,GOLFT/CF(K5),ON,CE(K5)                                      SP 650
        TLON,GOLFT/CE(K5),ON,CF(K5+1)                                    SP 660
        TLON,GOLFT/CF(K5+1),ON,CO(K5)                                    SP 670
        K6=0                                                             SP 680
S21)    TLON,GOLFT/CO(K5-K6),ON,CF(K5-K6-1)                              SP 690
        TLON,GOLFT/CF(K5-K6-1),ON,CE(K5-K6-1)                            SP 700
        TLON,GOLFT/CE(K5-K6-1),ON,CF(K5+K6+2)                            SP 710
        TLON,GOLFT/CF(K5+K6+2),ON,CO(K5-K6-1)                            SP 720

PPRINT  ***********************************************************
PPRINT
PPRINT  A SPIRAL PASS HAS JUST BEEN COMPLETED
PPRINT
PPRINT  ***********************************************************

        K6=K6+1                                                          SP 730
        IF (K5-(K6+1)) S20,S20,S21                                       SP 740
S20)    TLON,GOLFT/CO(1),ON,CF(1)                                        SP 750
        GODLTA/0,0,(B+A),FUP                                             SP 760

$$      FIND OUT WHETHER OR NOT THERE IS STILL MATERIAL LEFT ON SIDE 1   SP 765

S22)    IF(KS(1))S23,S23,S24                                             SP 770
S24)    CALL/SPIR6,JJ=1                                                  SP 780
        IF (K7-1) S23,S25,S25                                            SP 790
S25)    CALL/SPIR11                                                      SP 793
        CALL/SPIR9                                                       SP 797

PPRINT  ***********************************************************
PPRINT  SEMIROUGHING ON CHECKSURFACE 2
PPRINT  ***********************************************************

S23)    RAPID                                                            SP 800
        GO TO/(POINT/X1(2),Y1(2),A)                                      SP 810
        GODLTA/0,0,-(B+A),FDN                                            SP 820
        FEDRAT/FD                                                        SP 830
        GO TO/(POINT/X1(3),Y1(3),-B)                                     SP 840

$$      FIND OUT WHETHER OR NOT THERE IS STILL MATERIAL LEFT ON SIDE 3   SP 975

        IF (KS(3)) S29,S29,S30                                           SP 930
S30)    CALL/SPIR6,JJ=2                                                  SP 990
        IF (K7-1) S29,S31,S31                                            SP 1000
S31)    CALL/SPIR12                                                      SP 1003
        CALL/SPIR10                                                      SP 1007
        JUMPTO/S100                                                      SP 1010
S29)    GODLTA/0,0,(A+B),FUP                                             SP 1020
S100)   TERMAC                                                           SP 1030
        PRINT/0
```

1.2. Turning macros.

1.2.1. Barstock turning macro : ROUGH


```
ROUGH  = MACRO/K,MDP,FIN,CL,VEL,RAD,FD                                  120

$$ K      IS THE NUMBER OF SURFACES (NOT NECESSARILY CHECKSURFACES)     130
$$ MDP    IS THE MAXIMUM DEPTH OF CUT                                   140
$$ FIN    IS  THE FINISHING ALLOWANCE                                   150
$$  CL    IS THE CLEARANCE ON THE FACE                                  160
$$ VEL    IS THE CUTTING SPEED          (FT/MIN)                        170
$$ RAD    IS THE CUTTER RADIUS    (INCH)                                180
$$ FD     IS THE FEEDRATE  (INCH/MINUTE)                                190

          CUTTER/(2*RAD+2*FIN)                                          200
          FEDRAT/FD,IPM                                                 210
          I=1                                                           220
          FROM/SP                                                       230
          RAPID                                                         240
          GOTO/(POINT/(CL+RAD+FIN),-(RD(1)+RAD+FIN),0)                  250
          ONTCUT                                                        260
          GODLTA/10,0,0                                                 270
          GO/(LINE/PARLEL,LY,XLARGE,CL)                                 280
          CUT                                                           290

$$  DETERMINING THE PARAMETER M                                         300

2)        IF (KC(I)) 23,24,25                                           310
23)       OBTAIN,CIRCLE/CS(I),XC,,,,,,                                  320
          P(I)=POINT/XLARGE,INTOF,(LINE/PARLEL,LX,YSMALL,RD(I)),CS(I)   330
          OBTAIN,POINT/P(I),X,,                                         340
26)       M=0                                                           350
          JUMPTO/28                                                     360
25)       OBTAIN,CIRCLE/CS(I),XC,YC,,,,,RC                              370
          P(I)=POINT/XSMALL,INTOF,(LINE/PARLEL,LX,YSMALL,RD(I)),CS(I)   380
          OBTAIN,POINT/P(I),X,,                                         390
          IF(XC-CL) 27,27,26                                            400
27)       M=1                                                           410
          JUMPTO/28                                                     420
24)       OBTAIN,LINE/CS(I),XA,YA,,                                     430
          IF(XA) 35,36,35                                               440
36)       IF(KC(I-1))37,38,39                                           450
37)       P(I)=POINT/XLARGE,INTOF,CS(I),CS(I-1)                         460
          JUMPTO/40                                                     470
39)       P(I)=POINT/XSMALL,INTOF,CS(I),CS(I-1)                         480
          JUMPTO/40                                                     490
38)       P(I)=POINT/INTOF,CS(I),CS(I-1)                                500
          JUMPTO/40                                                     510
35)       P(I)=POINT/        INTOF,(LINE/PARLEL,LX,YSMALL,RD(I)),CS(I)  520
40)       OBTAIN,POINT/P(I),X,,                                         530
          JUMPTO/26                                                     540


28)       IF( RD(I)-RD(I+1))1,5,3                                       550
1)        I=I                                                           560

PPOINT      WRONG MACRO, NO GROOVING POSSIBLE                           570

          JUMPTO/17                                                     580

$$  J    IS THE OPTIMUM NUMBER OF CUTS IF J WERE INTEGER                590

5)        I=I                                                           600

PPRINT    WE START MACHINING A NEW SECTION                              610

          J=(RD(I)-RD(I+1))/MDP                                         620

$$  DETERMINING THE NUMBER OF CUTS WE ARE GOING TO TAKE                 630
$$        I.E.  JJ=INT(J)+1  IF J IS NOT INTEGER                       640

          JJ=1                                                          650
11)       IF(J-JJ)12,12,10                                              660
10)       JJ=JJ+1                                                       670
          JUMPTO/11                                                     680
12)       DP=(RD(I)-RD(I+1))/JJ                                         690
```

```
$$  J1 IS A COUNTER FOR THE NUMBER OF CUTS IN THIS SECTION I
$$       ( IF J1 REACHES JJ THEN SECTION I IS DONE )

         J1=1
15)      RAPID
         B=RO(I+1)+OP*(JJ-J1)
         INDIRV/J,1,0
         GO/TO,(LINE/PARLEL,LX,YSMALL,B)
         A=VEL/2/3.1415926/5*12
         SPINDL/A,CLW

$$ IF M=0      : GOLEFT TO THE CHECKSURFACE
$$ IF M=1      : FIRST,GOLEFT TO THE CENTRE OF THE CIRCLE AND THEN
$$                   GO FORWARD TO THE CHECKSURFACE

         IF(M)      18,18,19

18)      TLLFT,GOLFT/(LINE/PARLEL,LX,YSMALL,B),TO,CS(I)
         JUMPTO/34


19)      TLLFT,GOLFT/(LINE/PARLEL,LX,YSMALL,B)$
         ,ON,(LINE/PARLEL,LY,XLARGE,XC)
         IF(J1-JJ)21,22,22
22)      IF (RO(I+1)+RC-ABSF(YC)) 33,33,21
21)      TLLFT,GOFWD/(LINE/PARLEL,LX,YSMALL,B),TO,CS(I)
         JUMPTO/34
33)      ONTCUT
         GODLTA/J,-10,0
         INDIRV/J,1,0
         GO/TO,(LINE/PARLEL,LX,YSMALL,RO(I+1))
         CUT

34)      IF(J1-1)  30,30,53
30)      IF (KC(I))  50,51,52
51)      IF (YA)  53,53,50
53)      TLLFT,GOLFT/CS(I),PAST,(LINE/PARLEL,LX,YSMALL,(B+OP))
         JUMPTO/56
50)      TLLFT,GOLFT/CS(I),TO,(LINE/PARLEL,LY,XSMALL,(-X))
         ONTCUT
         GODLTA/J,10,0
         GO/PAST,(LINE/PARLEL,LX,YSMALL,(B+OP))
         CUT
         JUMPTO/56
52)      IF(RO(I)+YC)  54,54,50
54)      TLLFT,GOLFT/CS(I),ON,(LINE/PARLEL,LX,YSMALL,(B+OP))
         GODLTA/J,-RAD,0
         ONTCUT
         GODLTA/J,10,0
         GO/PAST,(LINE/PARLEL,LX,YSMALL,(B+OP))
         CUT
56)      RAPID
         TLRGT,GOLFT/(LINE/PARLEL,LX,YSMALL,(B+OP)$
           ),PAST,(LINE/PARLEL,LY,XLARGE,CL)
20)      I=I
         IF(J1-JJ)13,14,14
13)      RAPID
         GODLTA/J,DP,0
         J1=J1+1
         JUMPTO/15
14)      IF(I-K)16,17,17
16)      RAPID
         GODLTA/J,DP,0
5)       I=I+1
         IF(I-K)2,2,17
17)      RAPID
         GO TO /SP
         TERMAC
```

1.2.2. Macro for machining forgings : SUPV1.

```
          SUPV1=MACRO/
$$        K1 IS THE NUMBER OF SHOULDERS
$$$       RAD IS THE RADIUS OF THE SHAFT (UNCUT)
$$$       FR(I) AND FA(I) :
$$$         RADIAL AND AXIAL DIMENSIONS OF SHOULDER I BEFORE MACHINING
$$$         (DIMENSIONS OF FORGING)
$$$       DR(I) AND DA(I) :
$$$         RADIAL AND AXIAL DIMENSIONS OF MACHINED SHOULDER
$$$$      FD IS THE FEEDRATE  (IPM)
$$$$      VEL IS THE SURFACE VELOCITY
$$$       CL IS THE CLEARANCE

$$        MDPF  MAXIMUM DEPTH OF CUT IN FACING
$$$       MDPR  MAXIMUM DEPTH OF CUT IN ROUGHING
$$        I IS COUNTING THE NUMBER OF SHOULDERS

          CLPRNT                                                          L 60
          CUTTER/(2*R)                                                    L 70
          SPINDL/VEL,SFM,CLW,RANGE,2                                      L 80
          COOLNT/ON

PPRINT    **************************************************************
PPRINT
PPRINT    THE CUTTER GOES TO A SAFE STARTING POSITION (0,-RAD,0)
PPRINT
PPRINT    **************************************************************

          RAPID                                                           L 250
          GO TO/(POINT/0,-RAD,0)                                          L 260
          I=0                                                             L 270
1)        I=I+1                                                           L 280
          IF (I-1) 9,9,7                                                  L 290
9)        A=RAD-FR(1)                                                     L 300
          JUMPTO/8                                                        L 310
7)        A=FR(I-1)-FR(I)                                                 L 320
8)        B=DA(I)-FA(I)                                                   L 330

          PPINT/3,A,B
          TRACUT/(MATRIX/TRANSL,-FA(I),-FR(I),0)                          L 340
          CALL/FACMAC,AA=A,BB=B                                           L 350
          TRACUT/NOMORE                                                   L 355
          IF (I-K1) 1,2,2                                                 L 360
2)        RAPID                                                           L 370
          GO TO/(POINT/-FA(K1),-FR(1),0)                                  L 380
          I=0                                                             L 390
STOP
PPRINT
PPRINT    **************************************************************
PPRINT
PPRINT    THE OPERATOR HAS TO CHANGE THE TOOL
PPRINT
PPRINT    **************************************************************
PPRINT
6)        I=I+1                                                           L 400
          IF (I-K1) 3,4,5                                                 L 410
3)        A=DA(I)-DA(I+1)                                                 L 420
          C=DA(I+1)                                                       L 425
10)       B=FR(I)-DR(I)                                                   L 430

          PRINT/3,A,B
          TRACUT/(MATRIX/TRANSL,-C,-DR(I),0)                              L 440
          CALL/FGHMAC,AA=A,BB=B                                           L 450
          TRACUT/NOMORE                                                   L 455
          JUMPTO/6                                                        L 456
4)        A=DA(I)                                                         L 460
          C=0                                                             L 465
          JUMPTO/10                                                       L 470
5)        COOLNT/OFF                                                      L 480
          SPINDL/OFF                                                      L 490
          TERMAC
        PPINT/0
```

```
        FACMAC=MACRO/AA,BB                                          FA1 10

$$      FACING MACRO                                                FA1 15

$$      H IS COUNTING THE NUMBER OF PASSES                          FA1 16
$$      I IS THE NUMBER OF PASSES TO BE TAKEN                       FA1 17

        J=0                                                         FA1 20
        DP=0                                                        FA1 30
F1)     J=J+1                                                       FA1 40
        DP=DP+MDPF                                                  FA1 50
        IF (DP-BB) F1,F2,F2                                         FA1 60
F2)     DP=BB/J                                                     FA1 70
        PRINT/3,J
        RAPID                                                       FA1 80
        GO TO/(PCINT/-DP,-(AA+CL),0)                                FA1 90
        H=0                                                         FA1 100
F5)     H=H+1                                                       FA1 110

PPRINT  **********************************************************
PPRINT
PPRINT  WE TAKE THE H TH PASS IN THIS FACING OPERATION
PPRINT
PPRINT  **********************************************************

        GODLTA/0,(AA+CL),0,FD                                       FA1 120
        GODLTA/DP,0,0,FD                                            FA1 130
        IF (H-J) F3,F6,F6                                           FA1 140
F3)     RAPID                                                       FA1 150
        GODLTA/0,-(AA+CL),0                                         FA1 160
        RAPID                                                       FA1 170
        GODLTA/-(2*DP),0,J                                          FA1 180
        JUMPTO/F5                                                   FA1 190
F6)     TERMAC                                                      FA1 200
        PRINT/0



        RGHMAC=MACRO/AA,BB                                          RM 10

$$      I IS THE NUMBER OF PASSES TO BE TAKEN                       RM 17

$$      ROUGHING MACRO                                              RM 15
$$      H IS COUNTING THE NUMBER OF PASSES                          RM 16

        J=0                                                         RM 20
        DP=0                                                        RM 30
R1)     J=J+1                                                       RM 40
        DP=DP+MDPR                                                  RM 50
        IF (DP-BB) R1,R2,R2                                         RM 60
R2)     DP=BB/J                                                     RM 70
        PRINT/3,J
        RAPID                                                       RM 80
        GO TO/(PCINT/CL,-(BB+DP),0)                                 RM 90
        RAPID                                                       RM 95
        GODLTA/0,(2*DP),J                                           RM 97
        H=0                                                         RM 100
R5)     H=H+1                                                       RM 110

PPRINT  **********************************************************
PPRINT
PPRINT  WE TAKE THE H TH PASS IN THIS ROUGHING OPERATION
PPRINT
PPRINT  **********************************************************

        GODLTA/-(CL+AA),0,J,FC                                      RM 120
        GODLTA/J,-DP,0,FC                                           RM 130
        RAPID                                                       RM 140
        GODLTA/(CL+AA),0,0                                          RM 150
        IF (H-J) R3,R6,R6                                           RM 160
R3)     RAPID                                                       RM 170
        GODLTA/0,(2*DP),0                                           RM 180
        JUMPTO/R5                                                   RM 190
R6)     TERMAC                                                      RM 200
        PRINT/J
```

2. Preprocessor (minicomputer) software.

```
APT0        BASIC V01B-02

10 REM MAIN PROGRAM
15 PRINT "PREPROCESSOR SOFTWARE PACKAGE ( PART PROGRAM PREPARATION )"
16 PRINT
20 GOSUB 1000
410 OPEN "DX1:PART1" AS FILE VF1$(100)=64
412 OPEN "DX1:COUNT" AS FILE VF6
413 VF6(1)=1\VF6(2)=1\VF6(3)=1
414 VF6(6)=X0\VF6(7)=Y0\VF6(8)=X1\VF6(9)=Y1
415 REM CALL SUBROUTINE 1500 TO INPUT THE GEOMETRICAL STATEMENTS
420 K=1\GOSUB 1500
422 VF6(4)=1\VF6(5)=K
424 GO TO 440
430 OPEN "DX1:PART1" AS FILE VF1$(100)=64
432 OPEN "DX1:COUNT " AS FILE VF6
440 IF VF6(4)=VF6(5) THEN 545
445 Y$="="\Z=POS(VF1(VF6(4)),Y$,1)\REM Z IS POS OF=
450 X$=SEG$(VF1(VF6(4)),Z+1,Z+1)
470 IF X$="C" THEN 530
480 IF X$="L" THEN 510
482 IF X$="P" THEN 490
484 GOSUB 1600
486 GO TO 440
490 CHAIN "DX1:POINT0"
510 CHAIN "DX1:LINE0"
530 CHAIN "DX1:CIRCL0"
545 OPEN "DX1:CANPNT" AS FILE VF3$(100)=8
550 OPEN "DX1:CANLIN" AS FILE VF4$(100)=8
560 OPEN "DX1:CANCIR" AS FILE VF5$(100)=8
600 PRINT \PRINT
601 PRINT "****************************************"
602 PRINT "GEOMETR. DEF. OF PART PROGR. (VERSION 1)"
603 PRINT "****************************************"
604 PRINT \PRINT
610 FOR K1=1 TO (VF6(1)-1)*4 STEP 4
612 A$=TRM$(VF3(K1))&"=POINT/"&TRM$(VF3(K1+1))&","
614 A$=A$&TRM$(VF3(K1+2))&","&TRM$(VF3(K1+3))
617 PRINT A$
620 NEXT K1
630 FOR K1=1 TO (VF6(2)-1)*5 STEP 5
640 B$=TRM$(VF4(K1))&"=LINE/CANON,"&TRM$(VF4(K1+1))&","
642 B$=B$&TRM$(VF4(K1+2))&",0,"&TRM$(VF4(K1+4))
646 PRINT B$
650 NEXT K1
660 FOR K1=1 TO (VF6(3)-1)*8 STEP 8
670 C$=TRM$(VF5(K1))&"=CIRCLE/CANON,"&TRM$(VF5(K1+1))
672 C$=C$&","&TRM$(VF5(K1+2))&",0,0,0,1,"&TRM$(VF5(K1+7))
680 PRINT C$
685 NEXT K1
690 PRINT "FINI"
700 PRINT \PRINT
701 PRINT "****************************************"
702 PRINT "GEOMETR. DEF. OF PART PROGR. (VERSION 2)"
703 PRINT "****************************************"
704 PRINT \PRINT
```

```
710 FOR K1=1 TO VF6(5)-1
720 PRINT TRM$(VF1(K1))
730 NEXT K1
740 PRINT "FINI"\PRINT \PRINT
750 GOSUB 2500 \IF K2=1 THEN 440
760 OPEN "DX1:LABEL" AS FILE VF7%(64)=8
765 K=0\J=0\I=0
768 IF VF6(1)<=1 THEN 805
770 FOR K=1 TO VF6(1)-1
780 VF7(2*K-1)=VF3(4*K-3)
790 VF7(2*K)="POINT"
800 NEXT K
805 IF VF6(2)<=1 THEN 845
810 FOR J=1 TO VF6(2)-1
820 VF7(2*K+2*J-1)=VF4(5*J-4)
830 VF7(2*K+2*J)="LINE"
840 NEXT J
845 IF VF6(3)<=1 THEN 885
850 FOR I=1 TO VF6(3)-1
860 VF7(2*J+2*K+2*I-1)=VF5(8*I-7)
870 VF7(2*J+2*K+2*I)="CIRCLE"
880 NEXT I
885 L=2*J+2*K+2*I
887 IF L=0 THEN 920
920 PRINT "DO YOU WANT TO INPUT THE VARIOUS INFORMATION WITH "
925 PRINT "          INTERACTIVE GRAPHICS (Y OR N)  ";
930 INPUT A$
935 IF A$="N" THEN 945
940 CHAIN "DX1:BEN1"
945 PRINT \PRINT "WHAT KIND OF OPERATION (MILLING(M) OR TURNING(T) ) ";
950 INPUT A$\PRINT
955 IF A$="T" THEN 965
960 CHAIN "DX1:MIL111"
965 CHAIN "DX1:APTURN"
970 END
1000 REM DISPLAY OF THE GEOMETRICAL APT STATEMENTS
1010 CALL "DFIX"(1500)
1020 CALL "INIT"
1030 PRINT "EXTREME VALUES OF THE SCREEN ARE X0,Y0,X1,Y1  "
1040 INPUT X0,Y0,X1,Y1\PRINT
1060 CALL "SCAL"(X0,Y0,X1,Y1)
1090 CALL "APNT"(X0,0,0,-5)
1100 CALL "VECT"(-X0+X1,0,0,0,0,2)
1110 CALL "APNT"(0,Y1,0,-5)
1120 CALL "VECT"(0,Y0-Y1,0,0,0,2)
1140 RETURN
1500 REM SUBROUTINE
1502 PRINT "INPUT ALL THE GEOMETRICAL STATEMENTS"
1504 PRINT "      LAST ONE IS : END"
1510 INPUT M$\VF1(K)=M$\IF M$="END" THEN 1540
1520 K=K+1
1530 GO TO 1510
1540 RETURN
1600 REM SUBROUTINE TO ESTABLISH THE NON DISPLAYED GEOMETRY
1610 A$=SEG$(VF1(VF6(4)),1,Z)
1620 B$=SEG$(VF1(VF6(4)),Z+2,64)
1630 VF1(VF6(4))=A$&B$
1640 VF6(4)=VF6(4)+1
1650 RETURN
```

```
2500 REM SUBROUTINE FOR FINAL CORRECTIONS
2510 PRINT "ARE YOU SATISFIED WITH THE GEOMETRY,(Y OR N) ";
2530 INPUT M$
2540 IF M$="Y" THEN 2780
2550 PRINT "HOW MANY GEOMETRICAL STATEMENTS DO YOU WANT TO MODIFY ?"
2560 INPUT M
2570 PRINT "INPUT THE NEW GEOMETRICAL STATEMENTS"
2580 K3=10000\Y$="="
2590 FOR K2=1 TO M
2600 INPUT C$
2610 Z=POS(C$,Y$,1)
2620 A$=TRM$(SEG$(C$,1,Z-1))
2630 FOR K1=1 TO VF6(5)-1
2640 M$=VF1(K1)\Z=POS(M$,Y$,1)
2650 B$=TRM$(SEG$(M$,1,Z-1))
2660 IF A$=B$ THEN 2680
2670 NEXT K1\PRINT "NO SUCH GEOMETRICAL STATEMENTS FOUND"\STOP
2680 VF1(K1)=C$
2690 IF K1>K3 THEN 2710
2700 K3=K1
2710 NEXT K2
2720 FOR K2=K3 TO VF6(5)-1
2730 CALL "ERAS"(K2)
2740 NEXT K2
2745 VF6(4)=K3\GOSUB 5000
2760 K2=1
2770 RETURN
2780 K2=-1
2790 RETURN
5000 REM SUBROUTINE
5010 I1=1\I2=1\I3=1\Y$="="
5020 FOR K1=VF6(4) TO VF6(5)
5030 Z=POS(VF1(K1),Y$,1)
5040 A$=TRM$(SEG$(VF1(K1),1,Z-1))
5050 IF I1=0 THEN 5090
5060 FOR J1=1 TO VF6(1)*4-3 STEP 4
5070 IF VF3(J1)=A$ THEN 5180
5080 NEXT J1
5090 IF I2=0 THEN 5130
5100 FOR J1=1 TO VF6(2)*5-4 STEP 5
5110 IF VF4(J1)=A$ THEN 5190
5120 NEXT J1
5130 IF I3=0 THEN 5170
5140 FOR J1=1 TO VF6(3)*8-7 STEP 8
5150 IF VF5(J1)=A$ THEN 5200
5160 NEXT J1
5170 GO TO 5210
5180 VF6(1)=(J1+3)/4\I1=0\GO TO 5210
5190 VF6(2)=(J1+4)/5\I2=0\GO TO 5210
5200 VF6(3)=(J1+7)/8\I3=0\GO TO 5210
5210 IF I1+I2+I3=0 THEN 5230
5220 NEXT K1
5230 RETURN
```

```
POINTO     BASIC V01B-02

20 OPEN "DX1:PART1" AS FILE VF1$(16)=64
30 OPEN "DX1:CANPNT" AS FILE VF3$(100)=8
40 OPEN "DX1:CANLIN" AS FILE VF4$(100)=8
50 OPEN "DX1:COUNT" AS FILE VF6
60 Y$="="
65 M$=VF1(VF6(4))
67 CLOSE VF1
70 Z=POS(M$,Y$,1)
80 A$=TRM$(SEG$(M$,1,Z-1))
90 VF3(4*VF6(1)-3)=A$
100 Y$="/"\Z$=","\GOSUB 2000
110 IF X$="INTOF" THEN 200
120 VF3(4*VF6(1)-2)=X$
130 Y$=","\GOSUB 2000
140 VF3(4*VF6(1)-1)=X$
150 GOSUB 2100
160 VF3(4*VF6(1))=X$
170 X=VAL(VF3(4*VF6(1)-2))
180 Y=VAL(VF3(4*VF6(1)-1))
190 GO TO 465
200 REM INPUT A POINT IN THE INTOF FORMAT
210 Y$=","\Z$=","\Y=Z+1\GOSUB 2000
220 FOR K1=1 TO VF6(2)-1
230 IF VF4(K1*5-4)=X$ THEN 260
240 NEXT K1
250 PRINT "NO SUCH A LINE NAME WAS FOUND"\STOP
260 A=K1\GOSUB 2100
270 FOR K1=1 TO VF6(2)-1
290 IF VF4(K1*5-4)=X$ THEN 320
300 NEXT K1
310 PRINT " NO SUCH A LINE NAME WAS FOUND"\STOP
320 B=K1
330 REM CALCULATION OF THE INTERSECTION OF TWO LINES
340 REM          USING CRAMER'S RULE
350 D=VAL(VF4(A*5-3))*VAL(VF4(B*5-2))
360 D=D-VAL(VF4(B*5-3))*VAL(VF4(A*5-2))
370 IF D<>0 THEN 390
380 PRINT "LINES DO NOT INTERSECT"\STOP
390 Z=VAL(VF4(A*5))*VAL(VF4(B*5-2))
400 Z=Z-VAL(VF4(B*5))*VAL(VF4(A*5-2))
410 X=Z/D
420 VF3(4*VF6(1)-2)=STR$(X)
430 Z=VAL(VF4(A*5-3))*VAL(VF4(B*5))
440 Z=Z-VAL(VF4(B*5-3))*VAL(VF4(A*5))
450 Y=Z/D
460 VF3(4*VF6(1)-1)=STR$(Y)
462 VF3(4*VF6(1))="0"
465 CALL "SUBP"(VF6(4))
470 CALL "APNT"(X,Y,0,0,1)
480 CALL "APNT"(X,Y,0,-5,-1)
490 CALL "ESUB"
550 VF6(1)=VF6(1)+1
555 VF6(4)=VF6(4)+1
570 CHAIN "DX1:APTO" LINE 430
580 END
```

```
2000 REM STRING MANIPULATIONS 1
2010 Y=POS(M$,Y$,Z)
2020 Z=POS(M$,Z$,Y+1)
2030 X$=SEG$(M$,Y+1,Z-1)
2040 X$=TRM$(X$)
2060 RETURN
2100 REM STRING MANIPULATIONS 2
2110 Y=POS(M$,Y$,Z)
2120 X$=SEG$(M$,Y+1,LEN(M$))
2130 X$=TRM$(X$)
2140 RETURN
```

LINE0 BASIC V01B-02

```
20 OPEN "DX1:PART1" AS FILE VF1$(16)=64
40 OPEN "DX1:CANPNT" AS FILE VF3$(100)=8
50 OPEN "DX1:CANLIN" AS FILE VF4$(100)=8
55 OPEN "DX1:COUNT" AS FILE VF6
57 X0=VF6(6)\Y0=VF6(7)\X1=VF6(8)\Y1=VF6(9)
60 Y$="="
65 M$=VF1(VF6(4))
70 Z=POS(M$,Y$,1)
80 A$=TRM$(SEG$(M$,1,Z-1))
85 J=5*VF6(2)-4
90 VF4(J)=A$
100 Y$="/"\Z$=","\GOSUB 2000
110 IF X$<>"CANON" THEN 200
120 Y$=","
130 FOR K1=1 TO 3
140 GOSUB 2000
150 VF4(J+K1)=X$
160 NEXT K1
170 GOSUB 2100
180 VF4(J+4)=X$
190 GO TO 580
200 FOR K1=1 TO VF6(1)-1
210 IF VF3(4*K1-3)=X$ THEN 240
220 NEXT K1
230 PRINT "NO SUCH POINT FOUND"\STOP
240 Y$=","\GOSUB 2100
250 Y$=SEG$(X$,1,6)
260 IF Y$="ATANGL" THEN 480
270 A=K1
280 FOR K1=1 TO VF6(1)-1
290 IF VF3(4*K1-3)=X$ THEN 320
300 NEXT K1
310 PRINT "NO SUCH POINT FOUND"\STOP
320 B=K1
322 U=4*A-3
324 V=4*B-3
330 D=VAL(VF3(U+1))-VAL(VF3(V+1))
340 IF D<>0 THEN 400
350 VF4(J+1)="1"
360 VF4(J+2)="0"
370 VF4(J+3)="0"
380 VF4(J+4)=VF3(U+1)
390 GO TO 580
400 E=(VAL(VF3(U+2))-VAL(VF3(V+2)))/D
410 F=(VAL(VF3(U+1))*VAL(VF3(V+2)))/D
420 F=F-(VAL(VF3(V+1))*VAL(VF3(U+2)))/D
430 D=SQR(1+E^2)
```

```
...      J+1
450 VF4(J+2)=STR$(1/D)
460 VF4(J+4)=STR$(I/D)
470 GO TO 580
480 Y$=",":Z=7:M$=X$
490 GOSUB 2100
500 X=VAL(X$)
501 IF ABS(X)=90 THEN 504
502 IF ABS(X)<>270 THEN 510
504 VF4(J+1)="1":VF4(J+2)="0":VF4(J+3)="0":VF4(J+4)=VF3(4*K1-2)
506 GO TO 580
510 IF ABS(X)<>180 THEN 516
514 M=0
515 GO TO 520
516 M=X*3.14159/180:M=SIN(M)/COS(M)
520 D=SQR(1+M^2)
530 VF4(J+1)=STR$(-M/D)
540 VF4(J+2)=STR$(1/D)
550 VF4(J+3)="0"
560 K2=VAL(VF4(J+1))*VAL(VF3(4*K1-2))+VAL(VF4(J+2))*VAL(VF3(4*K1-1))
566 IF ABS(K2)>1.00000E-03 THEN 570
567 VF4(J+4)="0"
568 GO TO 580
570 VF4(J+4)=STR$(K2)
580 IF VAL(VF4(J+2))<>0 THEN 640
590 U0=VAL(VF4(J+4))/VAL(VF4(J+1))
600 U1=U0
610 V0=Y0
620 V1=Y1
630 GO TO 665
640 M=-VAL(VF4(J+1))/VAL(VF4(J+2))
650 T=VAL(VF4(J+4))/VAL(VF4(J+2))
660 GOSUB 5000
665 CALL "SUBP"(VF6(4))
670 CALL "APNT"(U0,V0,1,-5)
680 CALL "VECT"(U1-U0,V1-V0,0,0,0,1)
690 CALL "ESUB"
740 VF6(2)=VF6(2)+1
745 VF6(4)=VF6(4)+1
770 CHAIN "DX1:APTO" LINE 430
780 END
2000 REM STRING MANIPULATIONS 1
2010 Y=POS(M$,Y$,Z)
2020 Z=POS(M$,Z$,Y+1)
2030 X$=SEG$(M$,Y+1,Z-1)
2040 X$=TRM$(X$)
2060 RETURN
2100 REM STRING MANIPULATIONS 2
2110 Y=POS(M$,Y$,Z)
2120 X$=SEG$(M$,Y+1,LEN(M$))
2130 X$=TRM$(X$)
2140 RETURN
5000 REM CALCULATE THE INTERSECTION OF LINES WITH THE SCREEN
5010 U0=X0
5020 U1=X1
5050 V0=M*X0+T
5060 V1=M*X1+T
5065 IF M=0 THEN 5170
5070 IF M<0 THEN 5130
5080 IF V0>=Y0 THEN 5100
5090 U0=(Y0-T)/M
5095 V0=Y0
```

```
5100 IF V1=Y1 THEN 5170
5110 U1=(Y1-T)/M
5115 V1=Y1
5120 GO TO 5170
5130 IF V0=Y1 THEN 5150
5140 U0=(Y1-T)/M
5145 V0=Y1
5150 IF V1=Y0 THEN 5170
5160 U1=(Y0-T)/M
5165 V1=Y0
5170 RETURN
```

CIRCLO BASIC V01B-02

```
10 DIM N(2),G(2),H(2)
20 OPEN "DX1:PART1" AS FILE VF1$(16)=64
30 OPEN "DX1:CANLIN" AS FILE VF4$(100)=8
40 OPEN "DX1:CANCIR" AS FILE VF5$(100)=8
50 OPEN "DX1:COUNT" AS FILE VF6
60 Y$="="
70 M$=VF1(VF6(4))
80 Z=POS(M$,Y$,1)
90 A$=TRM$(SEG$(M$,1,Z-1))
100 K=8*VF6(3)-7
110 VF5(K)=A$
120 Y$="/"\Z$=","\GOSUB 2000
130 IF X$<>"CANON" THEN 210
140 Y$=","
150 FOR K1=1 TO 6
160 GOSUB 2000 \VF5(K+K1)=X$
170 NEXT K1
180 GOSUB 2100 \VF5(K+7)=X$
190 X=VAL(VF5(K+1))\Y=VAL(VF5(K+2))
200 GO TO 965
210 K1=1\GOSUB 2200
220 Y$=","\GOSUB 2000
230 FOR K1=1 TO VF6(2)-1
240 IF VF4(5*K1-4)=X$ THEN 270
250 NEXT K1
260 PRINT "NO SUCH LINE NAME FOUND"\STOP
270 N(1)=K1*5-4
280 GOSUB 2000 \K1=2\GOSUB 2200
290 GOSUB 2000
300 FOR K1=1 TO VF6(2)-1
310 IF VF4(5*K1-4)=X$ THEN 340
320 NEXT K1
330 PRINT "NO SUCH LINE NAME WAS FOUND"\STOP
340 N(2)=K1*5-4
350 GOSUB 2000
360 IF X$="RADIUS" THEN 380
370 PRINT "WRONG APT STATEMENT"\STOP
380 GOSUB 2100
390 R=VAL(X$)
430 D=VAL(VF4(N(1)+1))*VAL(VF4(N(2)+2))
440 D=D-VAL(VF4(N(2)+1))*VAL(VF4(N(1)+2))
450 X1=(VAL(VF4(N(1)+4))*VAL(VF4(N(2)+2)))/D
460 X1=X1-(VAL(VF4(N(2)+4))*VAL(VF4(N(1)+2)))/D
470 Y1=(VAL(VF4(N(1)+1))*VAL(VF4(N(2)+4)))/D
480 Y1=Y1-(VAL(VF4(N(2)+1))*VAL(VF4(N(1)+4)))/D
490 IF VAL(VF4(N(1)+2))<>0 THEN 510
500 H(1)=3.14159/2\GO TO 520
510 H(1)=ATN(-VAL(VF4(N(1)+1))/VAL(VF4(N(1)+2)))
```

```
520 IF VAL(VF4(N(2)+2))<>0 THEN 540
530 H(2)=3.14159/2\GO TO 550
540 H(2)=ATN(-VAL(VF4(N(2)+1))/VAL(VF4(N(2)+2)))
550 IF H(1)<>H(2) THEN 570
560 PRINT "LINES DO NOT INTERSECT"\STOP
570 H1=(H(1)+H(2))/2\REM ABSOLUTE ANGLE OF LINE WITH XAXIS
580 IF H(1)>H(2) THEN 610
590 K1=G(1)\G(1)=G(2)\G(2)=K1
600 K1=H(1)\H(1)=H(2)\H(2)=K1
610 H2=(H(1)-H(2))/2\REM RELATIVE ANGLE BETWEEN LINES
620 IF H(2)<0 THEN 650 \GOSUB 9000
640 GO TO 810
650 IF H(1)<0 THEN 720
660 IF G(2)<>-2 THEN 680
670 G(2)=-1\GO TO 700
680 IF G(2)<>-1 THEN 700
690 G(2)=-2
700 GOSUB 9000
710 GO TO 810
720 IF G(1)<>-2 THEN 740
730 G(1)=-1\GO TO 760
740 IF G(1)<>-1 THEN 760
750 G(1)=-2
760 IF G(2)<>-2 THEN 780
770 G(2)=-1\GO TO 800
780 IF G(2)<>-1 THEN 800
790 G(2)=-2
800 GOSUB 9000
810 IF Z<>1 THEN 830
820 D=R/SIN(H2)\GO TO 880
830 IF Z<>2 THEN 850
840 D=R/COS(H2)\H1=H1+3.14159/2\GO TO 880
850 IF Z<>3 THEN 870
860 D=R/SIN(H2)\H1=H1+3.14159\GO TO 880
870 D=R/COS(H2)\H1=H1+1.5*3.14159
880 X=X1+D*COS(H1)
890 Y=Y1+D*SIN(H1)
900 VF5(K+1)=STR$(X)
910 VF5(K+2)=STR$(Y)
920 VF5(K+3)="0"
930 VF5(K+4)="0"
940 VF5(K+5)="0"
950 VF5(K+6)="1"
960 VF5(K+7)=STR$(R)
965 CALL "SUBP"(VF6(4))
970 FOR K1=1 TO 360 STEP 4
980 M=X+VAL(VF5(K+7))*COS((3.14159/180)*K1)
990 N=Y+VAL(VF5(K+7))*SIN((3.14159/180)*K1)
1000 CALL "APNT"(M,N,1)
1010 NEXT K1
1012 CALL "ESUB"
1020 VF6(3)=VF6(3)+1
1025 VF6(4)=VF6(4)+1
1040 CHAIN "DX1:APT0" LINE 430
1050 END
2000 REM STRING MANIPULATIONS 1
2010 Y=POS(M$,Y$,Z)
2020 Z=POS(M$,Z$,Y+1)
2030 X$=SEG$(M$,Y+1,Z-1)
2040 X$=TRM$(X$)
2060 RETURN
```

```
2100 REM STRING MANIPULATIONS 2
2110 Y=POS(M$,Y$,Z)
2120 X$=SEG$(M$,Y+1,LEN(M$))
2130 X$=TRM$(X$)
2140 RETURN
2200 REM IDENTIFICATION SUBROUTINE
2210 IF X$<>"XSMALL" THEN 2230
2220 G(K1)=-2\GO TO 2280
2230 IF X$<>"XLARGE" THEN 2250
2240 G(K1)=-1\GO TO 2280
2250 IF X$<>"YSMALL" THEN 2270
2260 G(K1)=1\GO TO 2280
2270 G(K1)=2
2280 RETURN
9000 REM SUBROUTINE TO FIND RELATIVE POSITION OF CIRCLE
9010 IF ABS(G(1))<>1 THEN 9050
9020 IF ABS(G(2))<>1 THEN 9040
9030 Z=4\GO TO 9080
9040 Z=1\GO TO 9080
9050 IF ABS(G(2))<>1 THEN 9070
9060 Z=3\GO TO 9080
9070 Z=2
9080 RETURN
MIL111    BASIC V01B-02

1 REM APT OBTAINS THE NECESSARY INPUT IN A NON GRAPHICAL INTERACTIVE
2 REM        WAY FOR A MILLING OPERATION (PART 1)
5 CALL "INIT"
10 OPEN "POCK" AS FILE VF1
20 OPEN "COUNT" AS FILE VF6
100 PRINT "NUMBER OF POCKETS ";
110 INPUT K\VF1(1)=K
115 I=0
120 I=I+1
125 VF6(11)=I
130 PRINT "DEPTH OF CUT IN POCKET ";I;
140 INPUT K\VF1(I*10+1)=K\REM B IN POCKET I
145 VF6(10)=K
150 PRINT "NUMBER OF CHECKSURFACES IN POCKET ";I;
160 INPUT K\VF1(I*10+2)=K\REM K1 IN POCKET I
170 PRINT "ROTATION ANGLE OF POCKET ";I;
180 INPUT K\VF1(I*10+3)=K\REM AG IN POCKET I
185 PRINT
190 CHAIN "DX1:OPMIL3"
200 OPEN "POCK" AS FILE VF1
205 OPEN "COUNT" AS FILE VF6
210 I=VF6(11)
215 IF I<>VF1(1) THEN 120
220 FOR I=1 TO VF1(1)
230 PRINT "WHICH CUTTERRADIUS (IN) DO YOU CHOOSE FOR POCKET ";I;
240 INPUT K\VF1(I*10+4)=K\REM R IN POCKET I
250 PRINT "WHICH FEED (IN/MIN) ";
260 INPUT K\VF1(I*10+5)=K\REM FD IN POCKET I
270 PRINT "WHICH VELOCITY (FT/MIN) ";
280 INPUT K\VF1(I*10+6)=K
290 VF1(I*10+6)=(VF1(I*10+6)/VF1(I*10+4))/(2*3.14159)\REM RP
295 VF1(I*10+6)=VF1(I*10+6)*12
300 NEXT I
310 PRINT \PRINT "GENERAL INFORMATION "\PRINT
320 PRINT "INNERTOLERANCE (IN) ";\INPUT K\VF1(2)=K\REM IT
330 PRINT "OUTERTOLERANCE (IN) ";\INPUT K\VF1(3)=K\REM OT
```

```
340 PRINT "FEED FOR PLUNGING (IN/MIN) ";
345 INPUT K\VF1(4)=K\REM FDW
350 PRINT "FEED FOR RETRACTING (IN/MIN) ";
355 INPUT K\VF1(5)=K\REM FUP
360 PRINT "RELIEVE HEIGHT (IN) ";\INPUT K\VF1(6)=K\REM A
370 PRINT "STARTING X COORDINATE (IN) ";
375 INPUT K\VF1(7)=K\REM C
380 PRINT "STARTING Y COORDINATE (IN) ";
385 INPUT K\VF1(8)=K\REM D
390 PRINT "STARTING Z COORDINATE (IN) ";
395 INPUT K\VF1(9)=K\REM E
400 CHAIN "DX1:MIL222"
410 END
```

```
MIL222      BASIC V01B-02

1 REM APT OBTAINS THE NECESSARY INPUT IN A NON GRAPHICAL INTERACTIVE
2 REM              WAY FOR A MILLING OPERATION (PART2)
10 OPEN "POCK" AS FILE VF1
20 OPEN "DX1:PART1" AS FILE VF2$(100)=64
30 OPEN "DX1:CANLIN" AS FILE VF4$(100)=8
40 OPEN "DX1:CANCIR" AS FILE VF5$(100)=8
60 OPEN "DX1:COUNT" AS FILE VF6
397 PRINT "WHICH MILLING METHOD (RGH1,RGH2 OR SPIRAL)  ";
398 INPUT A$\PRINT \K3=1
400 IF A$="SPIRAL" THEN 408
402 IF A$="RGH2" THEN 406
404 VF1(10)=3\GO TO 409
406 VF1(10)=4\GO TO 409
408 VF1(10)=5
409 GOSUB 2000
410 PRINT "INPUT YOUR IDENTIFICATION OR PARTNO STATEMENT ";
420 INPUT E$
425 PRINT
430 PRINT "8888888888888888888888888888 "
440 PRINT "    PART PROGRAM (POCKET) "
450 PRINT "8888888888888888888888888888 "
455 PRINT
460 PRINT "PARTNO ";E$
480 PRINT "MACHIN/CINAC1,300,CIRCUL,1,0 "
490 IF VF1(10)=5 THEN 570
500 PRINT "RESERV/CS,9,KS,9,N,9,G,9,F,9,DUM1,9,DUM2,9,DUM3,9,DUM4,9,$"
510 PRINT "X,9,Y,9,T1,9,T2,9,A1,9,A2,9,W,9,M,9,CA,9,D,9,$"
515 IF VF1(10)<>3 THEN 540
520 PRINT "Q,9,X1,9,Y1,9,X2,9,Y2,9,$"
525 PRINT "U,50,V,50"
530 GO TO 610
540 PRINT "Q,9,X1,9,Y1,9,X2,9,Y2,9"
550 GO TO 610
570 PRINT "RESERV/CS,4,KS,4,N,4,G,4,F,4,DUM1,4,DUM2,4,DUM3,4,DUM4,4,$"
580 PRINT "X,4,Y,4,T1,4,T2,4,A1,4,A2,4,W,4,M,4,CA,4,D,4,$"
590 PRINT "H,4,DIS,4,L,4,CB,4,CE,20,CF,20,CD,20,LA,20,Q,4,X1,4,$"
600 PRINT "Y1,4,X2,4,Y2,4"
610 PRINT "IT=";VF1(2)
620 PRINT "OT=";VF1(3)
630 PRINT "FDW=";VF1(4)
640 PRINT "FUP=";VF1(5)
650 PRINT "A=";VF1(6)
655 PRINT
660 PRINT "CALL/INITO,A=IT,B=OT"\PRINT
680 FOR I=1 TO VF1(1)
690 PRINT "B=";VF1(I*10+1)
```

```
700          .- ;VF1(I*10+2)
710 PRINT "AG=";VF1(I*10+3)
720 PRINT "R-";VF1(I*10+4)
730 PRINT "FD=";VF1(I*10+5)
740 PRINT "RF=";VF1(I*10+6)\PRINT
750 FOR J=1 TO VF1(I*10+2)
760 PRINT "KS(";J;")=";VF1((I-1)*10+J+100)
770 NEXT J\PRINT
780 FOR J=1 TO VF1(I*10+2)
790 PRINT "M(";J;")=";VF1((I-1)*10+J+200)
800 NEXT J\PRINT
810 FOR J=1 TO VF1(I*10+2)
820 PRINT "N(";J;")=";VF1((I-1)*10+J+300)
830 NEXT J\PRINT
840 PRINT "CALL/INIT1,A=AG"\PRINT
850 IF K3=2 THEN 890
855 Z$="="
860 FOR J=1 TO VF1(I*10+2)
865 C$=VF4(50+I*10+J)
870 GOSUB 4000 \Y$=TRM$(Y$)
875 PRINT "CS(";J;")=";Y$;"     $$";X$
880 NEXT J\GO TO 920
890 FOR J=1 TO VF1(I*10+2)
900 GOSUB 3000
910 NEXT J\PRINT
920 PRINT "CALL/INIT2,BB=K1"\PRINT
925 IF I<>1 THEN 940
930 PRINT "FROM/(POINT/";VF1(7);",";VF1(8);",";VF1(9);")"\PRINT
940 PRINT "TRACUT/MO"\PRINT
950 IF VF1(10)=5 THEN 1000
960 PRINT "CALL/FINMAC"
970 IF VF1(10)=4 THEN 990
980 PRINT "CALL/RGH1"\GO TO 1010
990 PRINT "CALL/RGH2"\GO TO 1010
1000 PRINT "CALL/SPIRAL"
1010 PRINT "PRINT/3,ALL"\PRINT
1020 PRINT "TRACUT/NOMORE"\PRINT
1030 PRINT \NEXT I
1050 PRINT "CALL/FINAL"\PRINT \PRINT "END"\PRINT "FINI"
1080 IF K3=2 THEN 1100
1090 K3=K3+1\PRINT \GO TO 430
1100 END
2000 REM SUBROUTINE IN WHICH THE USER DESCRIBES EVERY POCKET
2002 REM              BY CONTOURING IT
2010 PRINT \PRINT "EXAMPLE OF CONTOURING : GOFWD/L1,INTOF,YLARGE "
2020 FOR I=1 TO VF1(1)
2025 PRINT
2030 PRINT "CONTOUR POCKET ";I
2040 FOR J=1 TO VF1(I*10+2)
2050 X$="/"\Y$=","\Z$="="
2055 INPUT A$
2060 Z1=POS(A$,X$,1)\Z2=POS(A$,Y$,1)
2080 B$=SEG$(A$,1,Z1-1)
2090 IF Z2=0 THEN 2130
2095 C$=SEG$(A$,Z1+1,Z2-1)\Z2=POS(A$,Y$,Z2+1)
2110 D$=SEG$(A$,Z2+1,64)
2120 GO TO 2150
2130 C$=SEG$(A$,Z1+1,64)\D$=" "
2150 IF B$="GOBCK " THEN 2210
2160 IF B$="GOFWD " THEN 2200
2170 IF B$="GOLFT " THEN 2190
```

```
2180 VF1(10*(I-1)+J+200)=+2\GO TO 2220
2190 VF1(10*(I-1)+J+200)=+1\GO TO 2220
2200 VF1(10*(I-1)+J+200)=-1\GO TO 2220
2210 VF1(10*(I-1)+J+200)=-2\GO TO 2220
2220 GOSUB 4000 \VF4(50+I*10+J)=C$
2290 FOR K1=1 TO (VF6(2)-1)*5 STEP 5
2300 IF VF4(K1)=C$ THEN 2370
2305 NEXT K1
2310 FOR K1=1 TO (VF6(3)-1)*8 STEP 8
2315 IF VF5(K1)=C$ THEN 2340
2320 NEXT K1
2340 PRINT "IS THIS SURFACE CONCAVE (-1) OR CONVEX (+1) ";
2350 INPUT K
2360 VF1(10*(I-1)+J+100)=K\GO TO 2375
2370 VF1(10*(I-1)+J+100)=0
2375 A=10*(I-1)+J+1+300
2380 IF J<>VF1(I*10+2) THEN 2390 \A=10*(I-1)+301
2390 IF D$="XLARGE " THEN 2440
2400 IF D$="YSMALL " THEN 2450
2410 IF D$="YLARGE " THEN 2460
2415 IF D$="XSMALL" THEN 2430
2420 VF1(A)=0\GO TO 2470
2430 VF1(A)=-2\GO TO 2470
2440 VF1(A)=-1\GO TO 2470
2450 VF1(A)=+1\GO TO 2470
2460 VF1(A)=+2\GO TO 2470
2470 NEXT J
2480 NEXT I
2490 RETURN
3000 REM SUBROUTINE ESTABLISHING THE CANONICAL FORM OF A CIRCLE
3010 FOR K1=1 TO (VF6(3)-1)*8 STEP 8
3020 IF VF4(50+I*10+J)=VF5(K1) THEN 3040
3030 NEXT K1
3035 GO TO 3510
3040 PRINT "CS(";J;")=CIRCLE/CANON,";VF5(K1+1);",";VF5(K1+2);",0,0,0,1,";
3050 PRINT VF5(K1+7);"     $$ ";VF5(K1)
3060 GO TO 3550
3510 FOR K1=1 TO (VF6(2)-1)*5 STEP 5
3520 IF VF4(50+I*10+J)=VF4(K1) THEN 3540
3530 NEXT K1
3540 PRINT"CS(";J;")=LINE/CANON,";VF4(K1+1);",";VF4(K1+2);",0,";
3542 PRINT VF4(K1+4);"     $$ ";VF4(K1)
3550 RETURN
4000 FOR K1=1 TO VF6(5)-1
4010 Z=POS(VF2(K1),Z$,1)
4020 X$=SEG$(VF2(K1),1,Z-1)
4030 Y$=SEG$(VF2(K1),Z+1,64)
4040 IF C$=X$ THEN 4070
4050 NEXT K1
4060 PRINT "WRONG EXIT"\STOP
4070 RETURN
```

```
1 REM APT OBTAINS THE NECESSARY INPUT IN A NON GRAPHICAL INTERACTIVE WA
2 REM                    FOR A TURNING OPERATION
3 DIM C$(20)
430 OPEN "DX1:TURN" AS FILE VF1
450 PRINT "NUMBER OF SHOULDERS ";\INPUT K\VF1(1)=K
460 PRINT "INNER TOLERANCE (IN) ";\INPUT K\VF1(2)=K
462 PRINT "OUTER TOLERANCE (IN) ";\INPUT K\VF1(3)=K
464 PRINT "NOSE RADIUS OF TOOL (IN) ";\INPUT K\VF1(4)=K
466 PRINT "CUTTING VELOCITY (FT/MIN) ";\INPUT K\VF1(5)=K
470 PRINT "CLEARANCE (IN) ";\INPUT K\VF1(6)=K
480 PRINT "FEED (IN/MIN) ";\INPUT K\VF1(7)=K
490 PRINT "IS THE WORKPIECE A FORGING (UNCUT STEPPED SHAFT) (Y OR N)
492 INPUT A$
500 IF A$="N" THEN 550
510 VF1(10)=2\REM FORGING
520 PRINT "MAXIMUM DEPTH OF CUT IN ROUGHING OPERATION (IN) ";
522 INPUT K\VF1(8)=K
530 PRINT "MAXIMUM DEPTH OF CUT IN FACING OPERATION (IN) ";
532 INPUT K\VF1(9)=K
540 GO TO 600
550 VF1(10)=1\REM BAR STOCK
560 PRINT "FINISHING ALLOWANCE (IN) ";\INPUT K\VF1(8)=K
570 PRINT "MAXIMUM DEPTH OF CUT (IN) ";\INPUT K\VF1(9)=K
580 GO TO 600
590 OPEN "DX1:TURN" AS FILE VF1
600 PRINT "WHAT IS THE LARGEST RADIUS OF THE UNCUT SHAFT  ";
610 INPUT K\VF1(15)=K
620 PRINT "STARTING X COORDINATE (IN) ";
630 INPUT K\VF1(16)=K
640 PRINT "STARTING Y COORDINATE (IN) ";
650 INPUT K\VF1(17)=K
660 PRINT "STARTING Z COORDINATE (IN) ";
670 INPUT K\VF1(18)=K
680 PRINT \PRINT "DESCRIPTION OF THE DIFFERENT SHOULDERS "\PRINT
690 FOR I=1 TO VF1(1)
700 IF VF1(10)=2 THEN 760
710 PRINT "INPUT THE RD(";I;") VALUE ";
720 INPUT K\VF1(I*10+1)=K
730 PRINT "INPUT THE RC(";I;") VALUE ";
740 INPUT K\VF1(I*10+2)=K
750 GO TO 840
760 PRINT "INPUT THE FR(";I;") VALUE ";
770 INPUT K\VF1(I*10+1)=K
780 PRINT "INPUT THE DR(";I;") VALUE ";
790 INPUT K\VF1(I*10+2)=K
800 PRINT "INPUT THE FA(";I;") VALUE ";
810 INPUT K\VF1(I*10+3)=K
820 PRINT "INPUT THE DA(";I;") VALUE ";
830 INPUT K\VF1(I*10+4)=K
840 NEXT I
845 IF VF1(10)=2 THEN 985
850 OPEN "DX1:PART1" AS FILE VF2$(100)=64
855 OPEN "DX1:COUNT" AS FILE VF6
860 PRINT \PRINT "INPUT THE NAMES OF THE CHECKSURFACES "
870 PRINT "     THIS MUST BE DONE IN SUCCESSIVE ORDER "\PRINT
880 Y$="="
890 FOR I=1 TO VF1(1)
900 PRINT "INPUT THE CS(";I;") NAME ";
910 INPUT C$(I)
```

```
920 FOR K1 = 1 TO VF6(5)-1
930 Z=POS(VF2(K1)+\,1)
940 A$=SEG$(VF2(K1),1,Z-1)
945 Z=POS(VF2(K1),"(",1)+4)
950 IF C$(I)=A$ THEN 970
960 NEXT K1
965 PRINT "WRONG EXIT"\STOP
970 C$(I)="C$("+STR$(I,2)+")="+A$
975 C$(I)=TRM$(C$(I))+\"    $$ "+X$
980 NEXT I
985 PRINT
990 PRINT "INPUT YOUR IDENTIFICATION OR PARTPROGRAM STATEMENT "
992 INPUT C$(VF1(1)+1)
994 GOSUB 1000
995 END
1000 REM SUBROUTINE THAT PRINTS THE PART PROGRAM\PRINT
1005 PRINT
1010 PRINT "**************************"
1020 PRINT "        PART PROGRAM"
1030 PRINT "**************************"\PRINT
1040 PRINT "PARTNO "+C$(VF1(1)+1)
1050 IF VF1(10)=2 THEN 1280
1060 PRINT "MACHIN/LATHE,1 "
1070 PRINT "RESERV/CS,10,RD,10,KC,10,F,10 "
1080 PRINT \PRINT "$$ INPUT THE SYSTEM MACRO ROUGH "\PRINT
1090 PRINT "INTOL/ "+VF1(2)
1100 PRINT "OUTTOL/ "+VF1(3)
1110 PRINT "COOLNT/ON "\PRINT "CLPRNT"
1120 PRINT "SP=POINT/ "+VF1(16)+","+VF1(17)+","+VF1(18)
1130 FOR I=1 TO VF1(1)
1140 PRINT C$(I)
1150 NEXT I
1160 FOR I=1 TO VF1(1)
1170 PRINT "KC("+I+")="+VF1(I*10+2)
1180 NEXT I
1190 PRINT "RD(1)="+VF1(15)
1200 FOR I=1 TO VF1(1)
1210 PRINT "RD("+I+1+")="+VF1(I*10+1)
1215 NEXT I
1220 PRINT
1230 PRINT "CALL/ROUGH,K="+VF1(1)+",MDF="+VF1(9)+",FIN="+VF1(8)+"  $"
1235 PRINT "     ,CL="+VF1(6)+"  $"
1240 PRINT "     ,VEL="+VF1(5)+",RAD="+VF1(4)+",FD="+VF1(7)
1250 PRINT "PRINT/3,ALL "
1260 PRINT "END "\PRINT "FINI "
1270 GO TO 1490
1280 PRINT "MACHIN/LATHE,1,291,,15 "
1290 PRINT "RESERV/FR,20,FA,20,DR,20,DA,20 "
1300 PRINT \PRINT "$$ INPUT THE SYSTEM MACROS FACMAC,ROUMAC AND SUBVL "
1310 PRINT \PRINT "FD="+VF1(7)
1320 PRINT "CL="+VF1(6)
1330 PRINT "R="+VF1(4)
1340 PRINT "VEL="+VF1(5)
1350 PRINT "K1="+VF1(1)
1360 PRINT "RAD="+VF1(15)
1380 FOR I=1 TO VF1(1)
1390 PRINT "FR("+I+")="+VF1(I*10+1)
1392 PRINT "DR("+I+")="+VF1(I*10+2)
1394 PRINT "FA("+I+")="+VF1(I*10+3)
1396 PRINT "DA("+I+")="+VF1(I*10+4)
1398 PRINT
1400 NEXT I
```

```
1430 PRINT "MDEPT "$VF1(3)
1440 PRINT "MDEPS="$VF1(9)
1450 PRINT "FREQ/(POINT "$VF1(16)$","$VF1(17)"-"$VF1(18)$")"
1460 PRINT "CALL SUPV1
1470 PRINT "PRINT/3,ALL "
1480 PRINT "END "\PRINT "TIME "
1490 RETURN
```

```
OPMILL      BASIC V07B-02

10 REM MILLING OPTIMIZATION PROGRAM
20 REM   (CONNECTED TO THE MILL1 PROGRAM)
380 OPEN "COUNT" AS FILE #F6
390 A3=1
395 A2=22
400 DIM A(50)
405 FOR A1=1 TO A2
410 READ A(A1)
412 IF A(A1)=999 THEN 500
415 NEXT A1
430 DATA .1,.124,.125,.126,.2,.24,.25,.26,.3,.4,.49,.5,.51
435 DATA .6,.7,.8,.874,.875,.876,.9,1
440 DATA 999
500 K1=.3\REM-OVERHEAD COST ($/MIN)
505 K2=1\REM-TOOL COST CONSTANT($/TOOTH/IN)
510 K3=2\REM-TOOL CHANGE TIME(MIN)
515 K4=1.8*10^9\REM- TOOL LIFE EQUATION CONSTANT
520 K5=2\REM- VELOCITY EXPONENT
525 K6=1\REM- EQUIVALENT CHIP THICKNESS EXPONENT
530 K7=.5\REM- AXIAL DEPTH OF CUT EXPONENT
535 K8=6*10^5\REM- FORCE EQUATION CONSTANT
540 R0=.3\REM- RADIAL FORCE/TANGENTIAL FORCE
545 PRINT
550 R=A(A3)\W=2*R
600 D=VF6(10)
615 S1=8.00000E-03\S2=1.00000E-03\REM- MAX & MIN FEED (IN/TOOTH)
620 R1=1\R2=.09\REM- MAX & MIN TOOL RADII (INCH)
625 V1=1000\V2=10\REM- MAX & MIN PERP. VELOCITY (FPM)
630 S3=130000\REM- MAX STRESS ALLOWED (PSI)
635 P1=10\REM- MAX POWER AVAILABLE (H.P.)
640 V5=100\REM- THE CRITICAL SPEED (FPM)
645 T1=63025.4*P1/V5\REM- MAX TORQUE AVAILABLE (F-I)
700 IF D<0 THEN 720
710 PRINT "** INFEASIBLE DEPTH OF CUT.**"
715 GO TO 600
720 IF W<=R2 THEN 735
725 PRINT "** INFEASIBLE WIDTH, CHECK LOWER CUTTER RADIUS LIMIT."
730 GO TO 1760
735 IF W<=2*R1 THEN 790
740 PRINT "** INFEASIBLE WIDTH, CHECK UPPER CUTTER RADIUS LIMIT."
745 GO TO 1760
790 PRINT
800 PRINT "RADIUS";TAB(9)"FEED"TAB(24)"VEL."TAB(30)"COST/IN^2";
801 PRINT TAB(43)"TORQUE"TAB(51)"POWER"TAB(60)"STRESS"
802 PRINT "(IN)"TAB(9)"(IN/MIN)"TAB(24)"(FPM)"TAB(30)"($/IN^2)";
803 PRINT TAB(43)"(F-I)"TAB(51)"(H.P.)"TAB(60)"(PSI)"
804 R=A(A3)\W=2*R
865 Z=3.14159
870 X1=(2*3.14159)/Z
900 IF R<.125 THEN 930
905 IF R<.24999 THEN 935
910 IF R<.49999 THEN 940
915 IF R<.87499 THEN 945
920 IF R<=.875 THEN 950
```

```
930 N=2\GO TO 950
935 N=3\GO TO 950
940 N=4\GO TO 950
945 N=5\GO TO 950
950 R=R\GO TO 950
955 S0$=" "\T0$=" "
1000 IF N R THEN 1010
1005 S=S1\GO TO 1015
1010 S=S1/(SIN(Z))
1015 F0=(KS*D*N*S)^(8*3.14159)
1020 F1=(1-COS(2*Z))*R0*(2*Z-SIN(2*Z))
1025 F2=R0*(1-COS(2*Z))*(2*Z-SIN(2*Z))
1030 F=SQR((F0*F1)^2+(F0*F2)^2)
1035 L=A*R
1040 M=F*(L-0.2)
1045 T0=(K8*D*N*S*W)/(2*3.14159)
1050 S0=(2/(3.14159*R^3))*(M+SQR(M^2+T0^2))
1065 IF S0<=S3 THEN 1075
1070 S=S*(S3/S0)\S0=S3\S0$="$"\S1$=S0$\GO TO 1015
1075 IF T0<=T0 THEN 1500
1080 S=S*(T1/T0)\T0=T1\T0$="*"\T1$=T0$\GO TO 1015
1500 V=V2\D9=10\J9=0\P0$=" "
1505 GOSUB 5000
1510 Q=S*(Q/Z)^(K6)
1515 V3=12*V/(2*3.14159*R)
1520 F3=V3*N*S\C1=K2*N*R
1525 T6=(60000/V3)*(Z/(2*3.14159))
1530 T7=(60000/V3)*(1-Z/(2*3.14159))
1535 E9=(39*LOG(T7)-23*LOG(T6))/LOG(10)+37.5
1540 X9=E9*SQR(V3*X1)
1550 P0=(KS*D*V3*N*S*W)/(3.96*10^5)
1605 V3=12*V/(2*3.14159*R)
1610 F3=V3*N*S
1615 T9=(X1*K4)/(V^K5*Q^K6*D^K7*X9^2)
1620 U9=K1/F3+C1/(F3*T9)+(K1*K3)/(F3*T9)
1622 U9=U9/(2*R)
1625 V8=V+D9
1630 V3=12*V8/(2*3.14159*R)
1632 F3=V3*N*S
1635 T6=(60000/V3)*(Z/(2*3.14159))\T7=(60000/V3)*(1-Z/(2*3.14159))
1637 E8=(39*LOG(T7)-23*LOG(T6))/LOG(10)+37.5\X8=E8*SQR(V3*X1)
1640 T8=(X1*K4)/(V3^K5*Q^K6*D^K7*X8^2)
1645 U8=K1/F3+C1/(F3*T8)+(K1*K3)/(F3*T8)
1647 U8=U8/(2*R)
1650 IF U8<=U9 THEN 1655
1651 IF J9<>0 THEN 1700
1655 IF U8/U9>=1 THEN 1665
1660 V=V-2*D9\D9=D9+1\J9=J9+1\GO TO 1675
1665 IF U8/U9>=1 THEN 1675
1670 V=V+D9
1675 V3=12*V/(2*3.14159*R)
1680 P0=(K8*D*V3*N*S*W)/(3.96*10^5)
1690 GO TO 1515
1700 GOSUB 5500
1705 U0=U9\IF R>1 THEN 1900
1750 V=INT(V)\T0=INT(T0)\S0=INT(S0)\P0=(INT(P0*100)/100)
1755 PRINT USING(8)F3;TAB(25)V;TAB(29)U0;
1756 PRINT TAB(42)T0;TAB(50)P0;TAB(59)S0;P0$;T0$;S0$;T2$
1757 IF A$="END" THEN 1900
1760 A3=A3+1\R=A(A3)
1765 IF R<=1 THEN 1775
1767 IF R>1 THEN 1900
1770 IF R=999 THEN 1
```

```
1775 GO TO 804
1900 PRINT
1905 IF S1$=" " THEN 1907
1906 GO TO 1910
1907 PRINT " 1: FEED-LIMITED BY MAX. STRESS."
1910 IF P1$=" " THEN 1912
1911 GO TO 1915
1912 PRINT " 2:VELOCITY LIMITED BY MAX. POWER."
1915 IF T1$=" " THEN 1917
1916 GO TO 1925
1917 PRINT " 3: FEED LIMITED BY MAX. TORQUE."
1925 PRINT
2000 CHAIN "DX1:MILL11" LINE 200
2010 END
5000 REM- USING SIMPSON'S RULE TO FIND EQUIV CHIP THICK. Q
5100 A9=0\B9=2\N9=100
5110 H9=(B9-A9)/N9
5115 A8=0\B8=H9
5120 FOR I=1 TO 99 STEP 2
5125 A8=A8+((SIN(B8))^(1/K6))*4
5130 B8=B8+2*H9
5135 NEXT I
5140 B8=2*H9
5200 FOR I=2 TO 98 STEP 2
5210 A8=A8+((SIN(B8))^(1/K6))*2
5215 B8=B8+2*H9
5220 NEXT I
5225 Q=(H9/3)*(A8+(SIN(B9))^K6)
5230 RETURN.
5500 REM- SUBROUTINE TO TEST THE POWER CONSTRAINT.
5505 P0=((K8*D*N*S.W)/(3.96*10^5))*(12*V/(2*3.14159*R))
5510 IF P0<=P1 THEN 5570
5515 V=V*P1/P0
5520 V3=12*V/(2*3.14159*R)
5525 F3=V3*N*S
5530 T6=(60000/V3)*(Z/(2*3.14159))
5535 T7=(60000/V3)*(1-Z/(2*3.14159))
5540 E9=(39*LOG(T7)-23*LOG(T6))/LOG(10)+37.5
5545 X9=E9*SQR(V3*X1)
5550 T9=(X1*K4)/(V^K5*Q^K6*D^K7*X8^2)
5555 U9=K1/F3+C1/(F3*T9)+(K1*K3)/(F3*T9)
5557 U9=U9/(2*K)
5560 P0=P1\P0$="#"\P1$=P0$
5570 RETURN
5600 END
```

```
READY        BASIC V010-02

5 CALL "OPEN"(20)
10 CALL "INIT"
20 CALL "OPEN"(100,200,1,5)
30 A$="SPECIFY THE OPERATION"
40 CALL "TEXT"(A$)
50 CALL "APNT"(200,600,0,-5)
60 CALL "TEXT"("TURNING")
70 CALL "APNT"(200,400,0,-5)
80 CALL "TEXT"("MILLING")
90 CALL "APNT"(200,200,0,-5)
100 CALL "TEXT"("DRILLING")
110 FOR K=1 TO 3
120 CALL "CHOP"(K)
130 CALL "APNT"(400,K*200,0,-5)
140 CALL "VECT"(200,0)
150 CALL "ESUB"
```

```
160 [illegible]
170 CALL "LPEN"(H,T)
180 IF H=0 THEN 170
190 IF T=1 THEN 210
195 IF T=3 THEN 220
200 CHAIN "DK1:NEW2"\GO TO 230
210 PRINT "CONNECTION WITH OTHER PROGRAMS HAS NOT YET BEEN ESTABLISHED"
215 GO TO 600
220 OPEN "0:1:TURN" AS FILE #1
224 CALL "INIT"
225 CALL "APNT"(300,800,1,-5)
230 CALL "TEXT"("TYPE OF SHAFT ")
240 CALL "APNT"(200,600,0,-5)
250 CALL "TEXT"("BAR STOCK SHAFT ")
260 CALL "APNT"(200,300,0,-5)
270 CALL "TEXT"("FORGING (UNCUT STEPPED SHAFT) ")
280 CALL "SUBP"(7)
285 CALL "APNT"(400,300,0,-5)
290 CALL "VECT"(100,0)
295 CALL "ESUB"
300 CALL "SUBP"(8)
305 CALL "APNT"(700,500,0,-5)
310 CALL "VECT"(100,0)
320 CALL "ESUB"
330 T=0
340 CALL "LPEN"(H,T)
350 IF H=0 THEN 340
360 IF T=8 THEN 480
380 VF1(10)=2\REM STEPPED SHAFT
390 GOSUB 1000
400 GOSUB 2000
420 CALL "TEXT"("PRINT THE MAXIMUM DEPTH OF CUT ")
422 CALL "APNT"(400,300,0,-5)
424 CALL "TEXT"("DURING ROUGHING OPERATION (IN)")
430 INPUT K\VF1(8)=K
440 GOSUB 2000
450 CALL "TEXT"("PRINT THE MAXIMUM DEPTH OF CUT ")
452 CALL "APNT"(400,300,0,-5)
454 CALL "TEXT"("DURING FACING OPERATION (IN)")
460 INPUT K\VF1(9)=K
470 GO TO 590
480 VF1(10)=1\REM BAR STOCK
500 GOSUB 1000
510 GOSUB 2000
530 CALL "TEXT"("PRINT THE FINISHING ALLOWANCE (IN) ")
540 INPUT K\VF1(8)=K
550 GOSUB 2000
570 CALL "TEXT"("PRINT THE MAXIMUM DEPTH OF CUT (IN) ")
580 INPUT K\VF1(9)=K
590 CHAIN "DK1:OPTURN" LINE 590
600 END
700 CALL "TIME"(2460)
710 CALL "TIMR"(T)
720 IF T>0 THEN 710
730 RETURN
1000 REM SUBROUTINE
1010 GOSUB 2000
1020 CALL "TEXT"("PRINT THE NUMBERS OF SHOULDERS ")
1030 INPUT K\VF1(1)=K
1040 GOSUB 2000
1050 CALL "TEXT"("PRINT THE INNER TOLERANCE (IN) ")
1060 INPUT K\VF1(2)=K
1070 GOSUB 2000
```

```
1080 CALL "TEXT":"PRINT THE CUTTER DIAMETER (IN)"
1090 INPUT KV$(1)=K
1100 GOSUB 2000
1110 CALL "TEXT":"PRINT THE TOOLRADIUS (IN) "
1120 INPUT KV$(3) "
1130 GOSUB 2000
1140 CALL "TEXT":"PRINT THE CUTTING VELOCITY (FT/MIN) "
1150 INPUT KV$(2)=K
1160 GOSUB 2000
1170 CALL "TEXT":"PRINT THE CLEARANCE (.1) "
1180 INPUT KV$(4)=K
1190 GOSUB 2000
1200 CALL "TEXT":"PRINT THE FEED (IN/MIN) "
1210 INPUT KV$(6)=K
1220 RETURN
2000 REM SUBROUTINE
2010 CALL "INIT"
2020 CALL "APNT"(200,500,0,-5)
2030 RETURN
```

BEN2 BASIC V01B-02

```
5 DIM N(20)
9 CALL "DFIX"(750)
10 CALL "INIT"
20 CALL "APNT"(50,1000,1,-5)
25 A$="WHICH OF THE FOLLOWING TOOLRADII HAVE YOU GOT AVAILABLE?"
30 CALL "TEXT"(A$)
32 B$="                              (INCHES)"
34 CALL "APNT"(50,950,0,-5)
36 CALL "TEXT"(B$)
40 FOR I=0 TO 9
45 K=900-I*80
50 CALL "APNT"(100,K,0,-5)
60 CALL "TEXT"(STR$(I*.05+.05))
70 CALL "SUBP"(I+1)
80 CALL "APNT"(300,K,0,-5)
90 CALL "VECT"(100,0)
100 CALL "ESUB"
110 CALL "APNT"(600,K,0,-5)
120 CALL "TEXT"(STR$(I*.05+.55))
130 CALL "SUBP"(I+11)
140 CALL "APNT"(800,K,0,-5)
150 CALL "VECT"(100,0)
160 CALL "ESUB"
170 NEXT I
180 CALL "APNT"(400,100,0,-5)
190 CALL "TEXT"("ALL OF THEM")
200 CALL "SUBP"(21)
210 CALL "APNT"(700,100,0,-5)
220 CALL "VECT"(100,0)
230 CALL "ESUB"
240 CALL "APNT"(800,20,0,-5)
250 CALL "TEXT"("READY")
260 CALL "SUBP"(22)
270 CALL "APNT"(900,20,0,-5)
280 CALL "VECT"(100,0)
290 CALL "ESUB"
300 OPEN "DX1:RADIUS" AS FILE #F1%(20)
305 FOR K=1 TO 20\VE1(K)=0\NEXT K
310 CALL "LPEN"(H,I)
```

```
320 IF T=0 THEN 340
330 IF T=22 THEN 370
340 IF T<>21 THEN 360
350 FOR K=1 TO 20\VF1(K)=1\NEXT K\GO TO 370
360 VF1(T)=1\GO TO 310
365 CLOSE VF1
370 OPEN "DX1:PARAM" FOR INPUT AS FILE VF2(17)
372 OPEN "DX1:PARAM1" FOR OUTPUT AS FILE VF3(17)
375 J=1
380 CALL "INIT"
390 CALL "APNT"(50,1000,1,-5)
400 A$="DO YOU AGREE WITH THE FOLLOWING VALUES"
410 CALL "TEXT"(A$)
420 CALL "APNT"(100,950,0,-5)
430 A$=" OF THE MILLING PARAMETERS ?"
432 CALL "TEXT"(A$)
434 CALL "APNT"(700,880,0,-5)
436 CALL "TEXT"("YES")
438 CALL "APNT"(900,880,0,-5)
440 CALL "TEXT"("NO")
450 IF J<>1 THEN 750
460 CALL "APNT"(0,800,0,-5)
470 CALL "TEXT"("OVERHEAD COST ($/MIN)")
480 CALL "APNT"(0,700,0,-5)
490 CALL "TEXT"("TOOLCOST CONSTANT ($/TOOTH/IN)")
500 CALL "APNT"(0,600,0,-5)
510 CALL "TEXT"("TOOL CHANGING TIME (MIN)")
520 CALL "APNT"(0,500,0,-5)
530 CALL "TEXT"("TOOLLIFE EQUATION CONSTANT")
540 CALL "APNT"(0,400,0,-5)
550 CALL "TEXT"("VELOCITY EXPONENT")
560 CALL "APNT"(0,300,0,-5)
570 CALL "TEXT"("EQUIVALENT CHIPTHICKNESS EXPONENT")
580 CALL "APNT"(0,200,0,-5)
590 CALL "TEXT"("AXIAL DEPTH OF CUT EXPONENT")
600 CALL "APNT"(0,100,0,-5)
610 CALL "TEXT"("FORCE EQUATION CONSTANT")
620 FOR K=0 TO 7
630 CALL "APNT"(500,800-K*100,0,-5)
640 CALL "TEXT"(STR$(VF2(K+1)))
650 CALL "SUBP"(2*K+1)
660 CALL "APNT"(700,800-K*100,0,-5)
670 CALL "VECT"(100,0)
680 CALL "ESUB"
690 CALL "SUBP"(2*K+2)
700 CALL "APNT"(900,800-K*100,0,-5)
710 CALL "VECT"(100,0)
720 CALL "ESUB"
730 NEXT K
732 GOSUB 3000
735 I1=0\I2=7\GOSUB 2000
740 J=J+1\GO TO 380
750 CALL "APNT"(0,800,0,-5)
760 CALL "TEXT"("MINIMUM FEED PER TOOTH (IN/TOOTH)")
770 CALL "APNT"(0,710,0,-5)
780 CALL "TEXT"("MAXIMUM FEED PER TOOTH(IN/TOOTH)")
790 CALL "APNT"(0,620,0,-5)
800 CALL "TEXT"("MINIMUM CUTTERRADIUS (INCH)")
810 CALL "APNT"(0,530,0,-5)
820 CALL "TEXT"("MAXIMUM CUTTERRADIUS (INCH)")
830 CALL "APNT"(0,440,0,-5)
```

```
840 CALL "TEXT"("MINIMUM CUTTING SPEED (FT/MIN)")
850 CALL "APNT"(0,350,0,-5)
860 CALL "TEXT"("MAXIMUM CUTTING SPEED (FT/MIN)")
870 CALL "APNT"(0,260,0,-5)
880 CALL "TEXT"("MAXIMUM STRESS (PSI)")
890 CALL "APNT"(0,170,0,-5)
900 CALL "TEXT"("MAXIMUM POWER (HP)")
910 CALL "APNT"(0,80,0,-5)
920 CALL "TEXT"("MAXIMUM TORQUE (F-I)")
930 FOR K=0 TO 8
940 CALL "APNT"(300,800-K*90,0,-5)
950 CALL "TEXT"(STR$(VF2(K+9)))
960 CALL "SUBP"(2*K+17)
970 CALL "APNT"(700,800-K*90,0,-5)
980 CALL "VECT"(100,0)
990 CALL "ESUB"
1000 CALL "SUBP"(2*K+18)
1010 CALL "APNT"(900,800-K*90,0,-5)
1020 CALL "VECT"(100,0)
1030 CALL "ESUB"
1040 NEXT K
1042 GOSUB 3000
1050 I1=8\I2=16\GOSUB 2000
1060 CLOSE VF2\CLOSE VF3
1070 CALL "INIT"
1080 CALL "APNT"(100,800,1,-5)
1090 A$="SPECIFICATION OF THE MILLING OPERATION"
1100 CALL "TEXT"(A$)
1110 CALL "APNT"(250,600,0,-5)
1120 CALL "TEXT"("PROFILING")
1130 CALL "APNT"(250,400,0,-5)
1140 CALL "TEXT"("POCKETING")
1150 CALL "SUBP"(1)
1160 CALL "APNT"(500,600,0,-5)
1170 CALL "VECT"(100,0)
1180 CALL "ESUB"
1190 CALL "SUBP"(2)
1200 CALL "APNT"(500,400,0,-5)
1210 CALL "VECT"(100,0)
1220 CALL "ESUB"
1230 CALL "LPEN"(H,T)
1240 IF H=0 THEN 1230
1250 IF T=2 THEN 1230
1260 PRINT "CONNECTION WITH PROFILING PROGRAM NOT ESTABLISHED YET"
1270 STOP
1280 CHAIN "DX1:REN3"
1290 END
2000 REM SUBROUTINE FOR LIGHTPEN INTERACTION
2010 FOR I=I1 TO I2\VF3(I+1)=VF2(I+1)\N(I+1)=0\NEXT I
2020 CALL "LPEN"(H,T)
2030 IF H=0 THEN 2020
2040 IF T>2*INT(T/2) THEN 2100
2050 N(T/2)=1
2060 GO TO 2110
2100 VF3((T+1)/2)=VF2((T+1)/2)
2105 N((T+1)/2)=0
2110 IF T<>40 THEN 2020
2120 FOR K=I1+1 TO I2+1
```

```
2130 PRINT \PRINT "PARAMETER ";K
2140 IF N(K)<>0 THEN 2170
2150 PRINT "      ";"VALUE REMAINS ";VF3(K)
2160 GO TO 2200
2170 PRINT "      ";"NEW VALUE  ";
2180 INPUT M
2190 VF3(K)=M
2200 NEXT K
2210 RETURN
3000 REM SUBROUTINE THAT DISPLAYS "READY" TAG
3010 CALL "APNT"(0,5,0,-5)
3020 CALL "TEXT"("READY")
3030 CALL "SUBP"(40)
3040 CALL "APNT"(100,5,0,-5)
3050 CALL "VECT"(100,0)
3060 CALL "ESUB"
3070 RETURN
```

```
DEN3      BASIC V01B-02

10 GOSUB 2000
15 CALL "INIT"
20 CALL "APNT"(100,900,1,-5)
30 CALL "TEXT"("HOW MANY POCKETS DO YOU WANT TO MILL  ?")
40 FOR K=1 TO 10
50 CALL "APNT"(300,810-K*80,0,-5)
60 A$=STR$(K)
70 CALL "TEXT"(A$)
80 CALL "SUBP"(K)
90 CALL "APNT"(500,810-K*80,0,-5)
100 CALL "VECT"(100,0)
110 CALL "ESUB"
115 NEXT K
120 CALL "LPEN"(H,T)
130 IF H=0 THEN 120
140 K2=T
150 PRINT K1,K2
160 END
2000 REM SUBROUTINE
2010 CALL "INIT"
2020 CALL "APNT"(0,1000,1,-5)
2030 CALL "TEXT"("THREE TYPES OF POCKETING:")
2040 CALL "APNT"(100,950,0,-5)
2045 A$="THE UNIDIRECTIONAL METHOD"
2050 CALL "TEXT"(A$)
2060 CALL "APNT"(100,900,0,-5)
2065 A$="THE ZIGZAG METHOD"
2070 CALL "TEXT"(A$)
2080 CALL "APNT"(100,850,0,-5)
2085 A$="THE SPIRAL METHOD"
2090 CALL "TEXT"(A$)
2100 CALL "APNT"(150,800,0,-5)
2105 A$="IN ORDER TO AVOID CUSPS,THE ROUGHING PASSES OVERLAP."
2110 CALL "TEXT"(A$)
2120 CALL "APNT"(150,750,0,-5)
2125 A$="BECAUSE OF THE COMPLEXITY INVOLVED IN MINIMISING OVERLAPPING,
```

```
2130 CALL "TEXT"(A$)
2140 CALL "APNT"(200,700,0,-5)
2150 A$="ONLY 'SIMPLE' RECTANGULAR TYPE OF POCKETS ARE ALLOWED."
2160 CALL "TEXT"(A$)
2170 CALL "APNT"(200,650,0,-5)
2180 A$="I.E. ONLY 4 OBJECT SURFACES OF WHICH 2 PARALLEL WITH AXIS."
2190 CALL "TEXT"(A$)
2200 CALL "APNT"(150,600,0,-5)
2210 A$="THE SPIRAL METHOD IS A 'BUILDING' TECHNIQUE."
2220 CALL "TEXT"(A$)
2230 CALL "APNT"(100,500,0,-5)
2240 CALL "TEXT"("WHICH MILLING METHOD WOULD YOU LIKE TO USE ?")
2250 CALL "APNT"(200,300,0,-5)
2260 CALL "TEXT"("ZIGZAG MILLING METHOD")
2262 CALL "APNT"(200,200,0,-5)
2263 CALL "TEXT"("UNIDIRECTIONAL METHOD")
2270 CALL "APNT"(200,100,0,-5)
2280 CALL "TEXT"("SPIRAL MILLING METHOD").
2290 CALL "SUBP"(1)
2300 CALL "APNT"(500,300,0,-5)
2310 CALL "VECT"(100,0)
2320 CALL "ESUB"
2322 CALL "SUBP"(2)
2324 CALL "APNT"(500,200,0,-5)
2326 CALL "VECT"(100,0)
2328 CALL "ESUB"
2330 CALL "SUBP"(3)
2340 CALL "APNT"(500,100,0,-5)
2350 CALL "VECT"(100,0)
2360 CALL "ESUB"
2370 CALL "LPEN"(H,T)
2380 IF H=0 THEN 2370
2390 K1=T
2400 CALL "TIME"(3*60)
2410 CALL "TIMR"(E)
2420 IF E<>0 THEN 2410
2430 RETURN

READY
```

Appendix 3 :

Auxiliary calculations concerning the SPIRAL macro.

Cusps along the cutterpath can be avoided by letting parallel cutterpaths overlap. The amount of overlapping required can be calculated by easy geometrical manipulations. Fig. A3.1 depicts the material left uncut when full slotting, while Fig. A3.2 represents the necessary variables in the calculation of the smallest overlapping (y_{lim}) in order for all cusps to vanish. The following equations can easily be gathered from Fig. A3.3.

$$\frac{x + y_{lim} + b}{b} = \frac{x}{y_{lim}} \quad \text{and} \quad b = \frac{y_{lim}}{\sin \frac{\alpha}{2}}$$

$$\frac{x + y_{lim} + \frac{y_{lim}}{\sin \frac{\alpha}{2}}}{\frac{y_{lim}}{\sin \frac{\alpha}{2}}} = \frac{x}{y_{lim}}$$

$$\sin \frac{\alpha}{2} (x + y_{lim}) + y_{lim} = x \quad \text{or} \quad y_{lim} = x \frac{1 - \sin \frac{\alpha}{2}}{\sin \frac{\alpha}{2}}$$

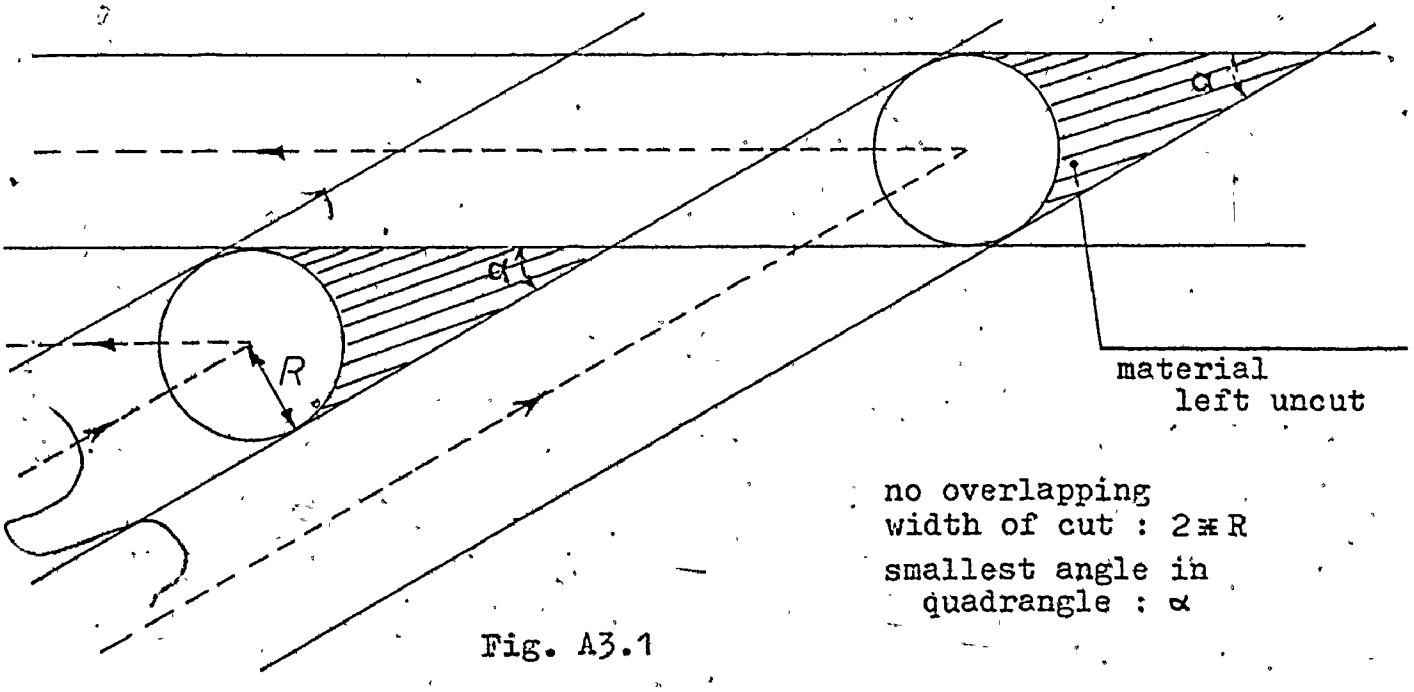
$$\text{thus} \quad x + y_{lim} = y_{lim} * \frac{1}{1 - \sin \frac{\alpha}{2}}$$

$$R = y_{lim} * \frac{1}{1 - \sin \frac{\alpha}{2}}$$

$$\text{or} \quad y_{lim} = R (1 - \sin \tfrac{\alpha}{2})$$

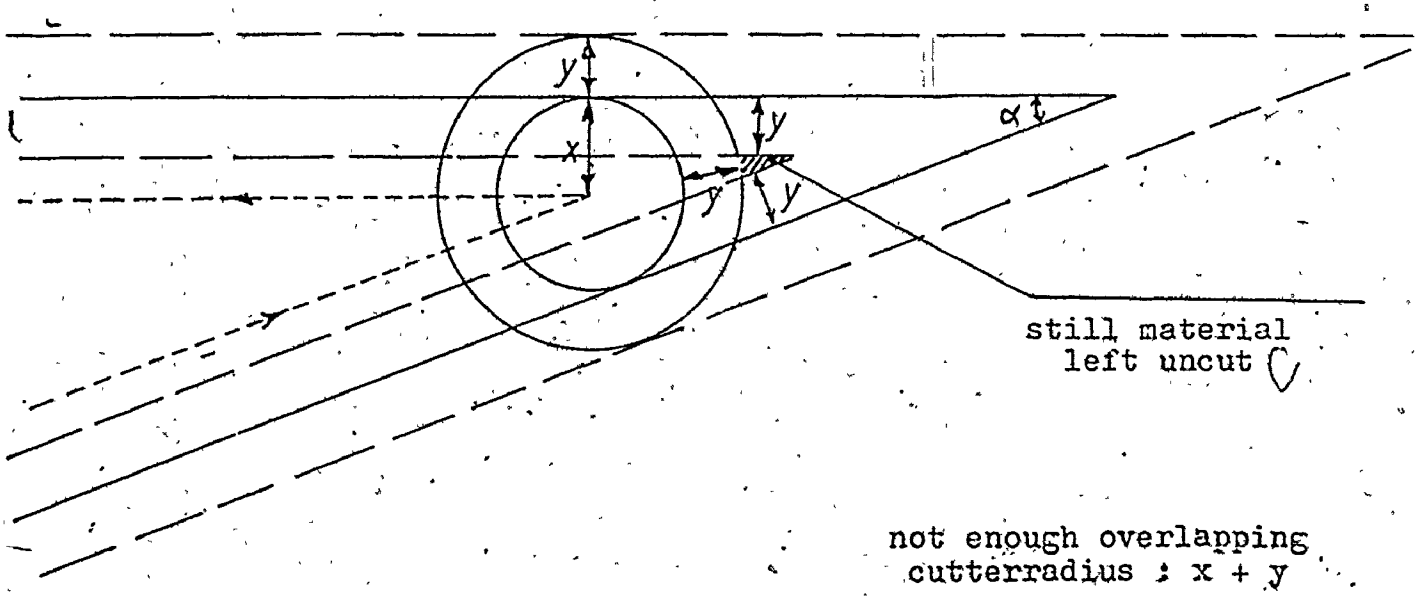
The width of cut is $w = 2 \times R - 2 \times y_{lim}$

$$w = 2 \times R \times \sin \frac{\alpha}{2}$$



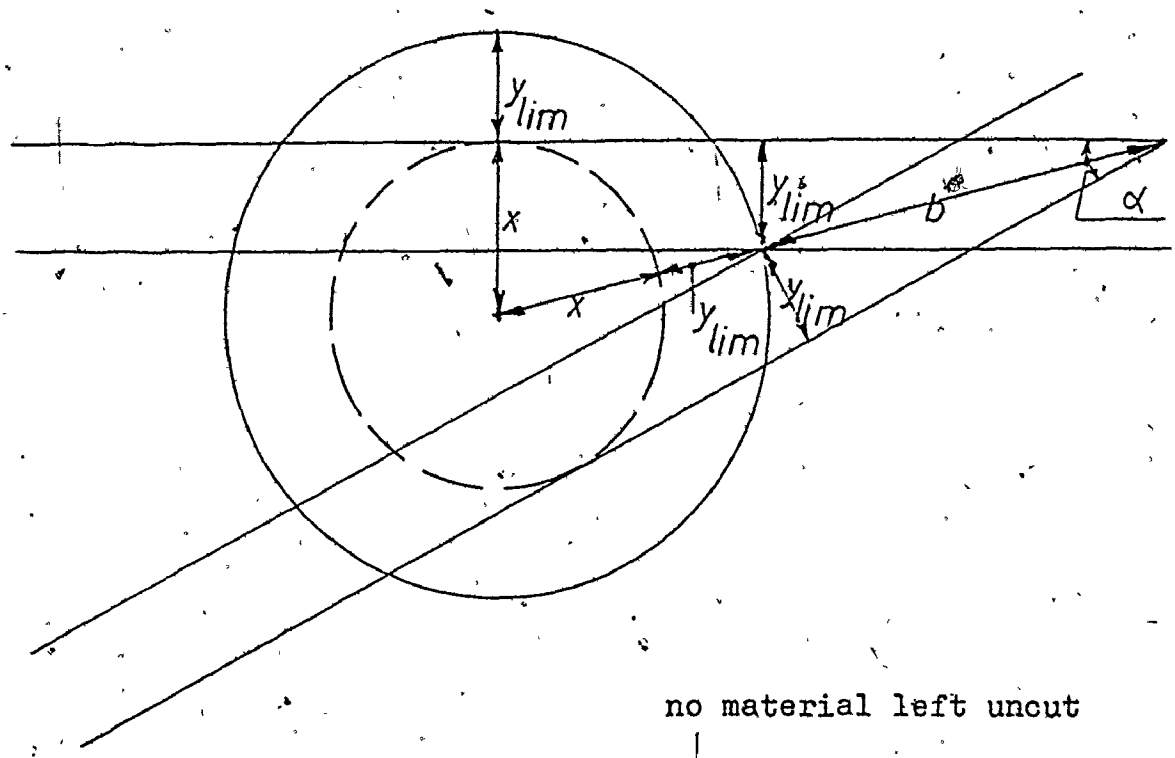
no overlapping
width of cut : $2 \times R$
smallest angle in
quadrangle : α

Fig. A3.1



not enough overlapping
cutterradius : $x + y$
width of cut : $2x$
smallest angle in
quadrangle : α

Fig. A3.2



no material left uncut

Fig. A3.3

Example :

While pocketing a rectangle with the SPIRAL module, the maximum of the ratio of the width of cut over the cutterdiameter ought to be :

$$\frac{w}{2*R} = \sin\frac{\alpha}{2} = \sin 45^\circ = \frac{\sqrt{2}}{2} \simeq .707$$