UNITED ARAB REPUBLIC

THE INSTITUTE OF NATIONAL PLANNING

التخطيط القومي

Memo. No. 893

NUMERICAL CONTROL SERIES
PROGRAMMING SYSTEMS

PART I : APT

BY

ALWALID ELSHAFEI
OPERATIONS RESEARCH GROUP

MAY 1969

CAIRO
3. MOHAMED MAZHAR - ZAMALEK

INTRODUCTION

In the first article of this series, we have quickly surveyed some of the available computer languages for programming NC equipment. As I have mentioned there, some languages are designed only for simple point-to-point work; others, for continuous-path operation. Some programmes are designed for specific equipment where the processor and postprocessor are contained within one programme; others are designed for general-purpose use where separate processor and postprocessor are used to allow the output to be used for any control system. There also are both general-purpose and special-purpose post-processor programmes. Some of these programmes are readily available, others are highly proprietary. Some programmes can be used on particular computers; others can be used on a variety of computers. Some programmes are even developed within a single company to meet special needs. Among this wide array, only a few are considered to be major languages. This article (which is composed of several parts) will handle some of those major languages. Abstractions from different industrial situations and research articles will be called upon for illustration.

This part, being the first of the article, will be dealing with APT.* APT is the most sophisticated of all NC programming systems. It is used for speeding the preparation of control tapes for NC m/c's through describing three-dimensional geometric configurations and tool motions on space curves. Up to six m/c Tool axes may be programmed. APT has also been designed for programming point-to-point work, straight line m/c'g, 2D and 2½ D contour m/c'g and lathe work.

---

* Automatic Programmed Tools

CONSTRUCTION OF THE APT SYSTEM

The APT SYSTEM consists of two main parts:

* The APT language, as a programming language
* The PROCESSOR; a number of computer programmes whose function is to interpret the language and compute the tool path.

The APT Language:

It consists of several hundreds of the English-like words, those words are used to transform different descriptive and instructive statements into a form acceptable by the computer, this is usually done by the part programmer.

The part programmer needs to have very little knowledge about the computer that uses the APT programmes to process the language. He should, by all means, devote his complete concentration on the production process to be programmed using the APT vocabulary. The construction of its free format statements is easily learned, and can be punched into a card for feeding into the computer without fixed locations.

To eliminate errors as far as possible, the part programmer only uses the coordinates, dimensions and so on, he finds in the engineering drawing. A part programme in the APT language can easily be read and understood.

The Processor

The processor establishes a general solution based on the part programme. It is obtained by geometric calculations and consists of general representation of the coordinates of the cutter path along the contour of the part. Attached with the processor there is a group of subroutines for special types of calculations, those subroutines will be discussed in detail later on. The general solution obtained thereof is valid for a milling machine, a flame cutter or a drafting machine. After the general solution is obtained, the processor finishes its task and the post-processor takes over.

The post-processor is another computer programme used to transform the general solution into a format adequate for a certain machine tool control system combination.

Back to the processor, the processor of the APT system is very large. It consists of 75,000 IBM-7090 words. It took about 100 man years to develop. A post-processor to tailor the result of the processor to a particular requirement may be developed within 3 to 9 months, depending on the controller and the machine tool. At the end of 1968 there were about 300 post-processors available for the APT III system.

APT CAPABILITIES:

APT capabilities depend upon:

1. The variety of geometric routines now built into the programme which enable the part programmer to define the 3D contouring operations needed in production of his workpieces.
2. The case with which the routines can be selected and applied.
3. The efficiency with which they will convert the part programmer's instructions into machine langauge

PART PROGRAMMING

A part programme produced with the APT language may consist of seven parts:

- Description of the part geometry
- Description of the shape of the tool to be used
- Data for tolerances
- Description of the tool motion
- Data for machine tool functions
- Formulas for numerical and trigonometrical calculations
- Statements for the programme logic.

These parts will be dealt with in turn.

Describing the part geometry

In the APT language, there are about 100 geometric definitions, among which there are 10 for the definition of points, 11 for the definition of lines and 10 for the definition of circles. Moreover it is possible to define planes, vectors, ellipses, hyperbolas, cylinders, spheres, matrices, patterns and quadratic surfaces.

The general format of a definition statement is:

Symbol = APT surface/modifiers. The statement means that the geometric element given as an APT surface is to be defined by using the modifiers and is to be stored under the symbol, that is, its name. Any symbol may be chosen. However, it is recommended to use a symbol which in one way or another refers to the meaning of the defined element or to its function. As an APT surface the words specified in , vocabulary for geometrical elements are to be used. The modifers can either be numbers or other APT words or previously defined elements. Examples for geometrical definitions are shown in Fig. 1.

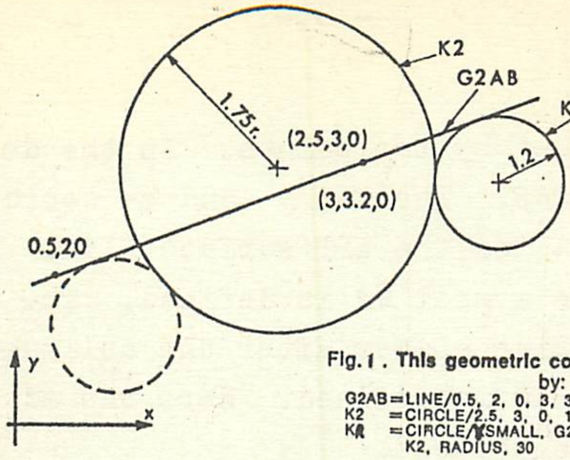
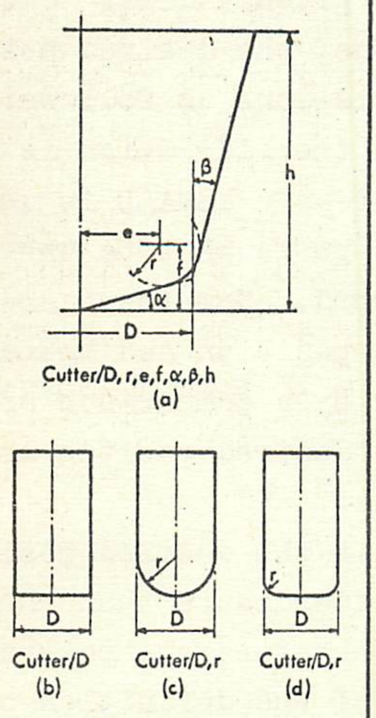


Fig. 1. This geometric configuration is defined by:

```
G2AB=LINE/0.5, 2, 0, 3, 3.2, 0                        1
K2  =CIRCLE/2.5, 3, 0, 1.75                           2
KR  =CIRCLE/YSMALL, G2AB, XLARGE, OUT, $
     K2, RADIUS, 30                                   3
```



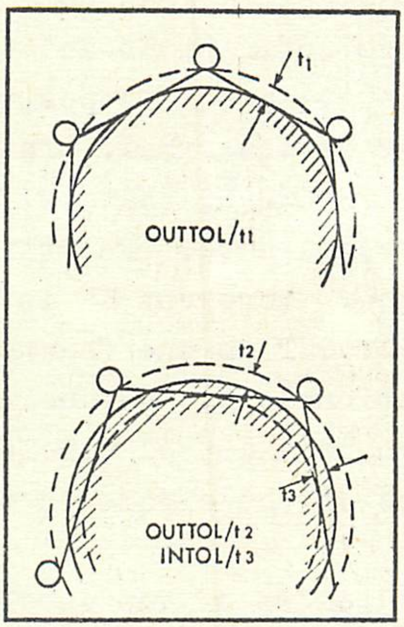


Fig. 2. Cutter definitions can incorporate seven modifiers (a), but simpler cutters (b, c, d) need only the first two.

Fig. 3. Permitted deviation from the programmed contour is indicated by the programmer using INTOL and OUTTOL statements.

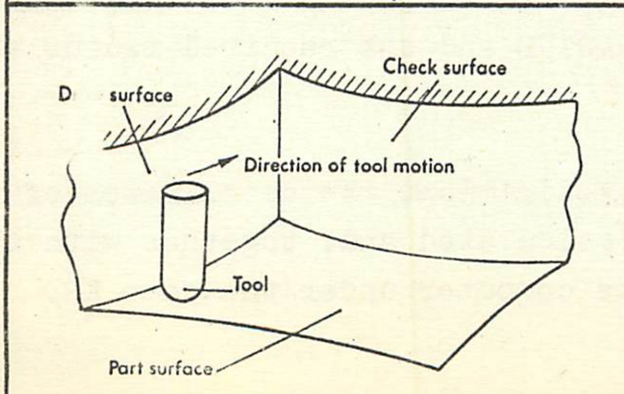


Fig. 4. Position of a tool in space is defined in the APT system by three surfaces: the drive surface, the part surface and the check surface.

A line is defined by two points. In the definition the line receives the symbol G2AB. The x-, y- and z- coordinates of both points follow the word for the APT surface LINE. In the next statement a circle with the symbol K2 is defined. The coordinates of the centre and the radius appear after the oblique line. In the third line the circle KR is defined. Here the advantages of a symbolic languages are obvious.

The circle KR can be "constructed" with reference to the two previous definitions and the given radius. Out of the eight (theoretically) possible positions, the desired position of the circle must be described. This is done as follows: the circle lies under the line G2AB, that is, on the side which is directed to the smaller y coordinate. For this reason YSMALL is used after the oblique line. Moreover KR is to be tangent to the circle K2 from the outside: OUT is the fourth word. Now there are only two possible positions left: the desired position and the undesired position, drawn as a dotted circle. By a comparison of the centre coordinates of the circle in the desired positon is determined.

The centre of the circle in the desired position has a larger x coordinate and a larger y coordinate than the centre of the circle in the other position. Therefore one puts the word XLARGE as a third word after the slash of the definition of KR. In this case, the word YLARGE would do just as well. At the end of the definition the word RADIUS and the required radius value are included.

Based on this definition, the coordinates of the centre of the circle KR can be calculated and, together with the radius, they are stored in the computer under the name KR.

A line or a circle in a drawing actually represent a plane or a cylinder which are perpendicular to the corresponding projection plane. For this reason in the APT system the line is considered as a plane perpendicular to the xy plane and the circle as a perpendicular cylinder. The part programmer, whenever he finds it of advantage, can programme a line of a circle. However, he must have a clear understanding of their interpretation by the APT system.

The tool shape

The APT system assumes rotational symetry of the tool. Fig.2 a shows the general shape of the tool. The statement consists of seven modifiers:

CUTTER/D, r, e, f, a, β, h

For the description of the profile a local coordinates system with its origin in the lowest point of the tool axis is used. The meaning of the first six parameters is explained by Fig. 2a. The parameter h, the height, has nothing to do with the actual length of a tool. This parameter represents the upper end of the profile, which, in 3D tool path calculations, is to be tested for contact with one of the three tool guiding surfaces. The test starts at the lowest point of the tool axis.

For simple tool profiles like those in Fig. 2 (b, c, d,) only the first or the first and second parameters are needed to define the tool.

Tolorancing data

For the previously described approximation of non-linear tool paths it is necessary that the part programmer indicates the permitted deviation from the programmed contour. With the data OUTTOL/t1 and INTOL/t2 a tolerance band is defined on both sides of the contour within which the point of tangency of the tool moves.

In a part programme both or only one of the values may be programmed. OUTTOL allows that at several places material remains outside the programmed contour, while through INTOL the programmed contour is undercut, Fig. 3.

When programming a part, one should programme a tolerance value which is one magnitude smaller than the tolerance of the machine tool. That means that more lines than necessary are produced. The post-processor will not consider unnecessary values.

Tool motions

To define the position of a tool in space, the APT system uses three surfaces, Fig. 4, which are:

a. Drive surface, abbreviated DS
b. Part surface, abbreviated PS
c. Check surface, abbreviated CS

The tool moves along the drive surface. In case of a vertical milling machine, for which the tool is perpendicular to the plane of the table, the xy plane, the drive surface corresponds to the contour and gives the xy coordinates of each point of the path of the tool centre.

The part surface is the surface on which moves the end point of the tool axis. For a vertical milling machine this surface gives the z coordinates of the respective path points.

The check surface terminates the motion of the tool moving along the element of the contour.

Each surface definable with the APT language may be assigned to be one of the three guiding surfaces.

When programming the tool motion a fixed part and a moving tool are assumed. In that case the part programmer acts as if he was sitting on the tool, driving it along the contour like a car on the road. The general structure of a motion statement is

Modifier 1, GO/drive surface, modifier 2, check surface.
The modifier 1 is sometimes required in complicated part programme to define the position of the tool in relation to the drive surface during the motion. The words Fig.5, to be used for that are:

TLRGT (tool right)
TLON (tool on)
TLLFT (tool left)

The second modifier, modifier 2, describes the position of the tool at the end of a motion in relation to the check surface. As modifier 2 one of the following words, Fig. 6, is used:

TO
ON
PAST
TANTO

The modifier to be used results from the starting position of the motion. For GO one of the words

GOFWD (go forward)
GORGT (go right)
GOLFT (go left)
GOBACK (go back)
GOUP (go up)
GODOWN (go down)

is to be inserted. They give the direction of the tool motion relative to the last motion of the tool.

Figure 7 demonstrates the application of these words. It is assumed that all geometric elements shown have been previously defined.

The part programme describing the tool motions always has to begin with the determination of the starting position of the tool. This is done in the statement FROM/.... Here (in line 1) the point PKT1 is the starting position, therefore: FROM/PKT1. The subsequents statement GO/TO, L2, ON, SRF in line 2 determines the drive surface (L2) and the part surface (SRF) for the following motion (in line 3). This motion is called "start up"

A linear motion over the shortest distance to L2 is assumed and the coordinates of the tool centre at position 1 are calculated. Then the tool is to move to position 2. This location is the point or rather the line of tangency between L2 and K1. It is the end of a motion along a part of the contour, that is along L2. Therefore, K1 is used as check surface and the statement describing this motion is GOLFT/L2, TANTO, K1, where the direction is related to the previously performed motion. TANTO indicates that the motion should be terminated at the point of tangency between drive surface and check surface.

In line 4 the motion along the circular arc is described. Since the first incremental move on the arc is in direction of the tangent line L2, GOFWD is used. The tool is to cross line L3, to cut a sharp corner. This is indicated by PAST in line 4 therefore PAST is used as modifier 2: GOFWD/K1, PAST, L3. The final position for this move is position 3. The last move ends in position 4, close to line L4. It is programmed by GORGT/L3, TO, L4.

To direct the end point of the tool axis to a previously determined point, the statement

GOTO/x, y, Z,

Or GOTO/P1

where P1 is a previously defined point, is to be used.

MACHINE TOOL FUNCTIONS

The part programmer may command the execution of machine tool functions between tool motion statements. For instance:

- determination of feedrates with

  FEDRAT/50

  FEDRAT/RAPID

- determination of spindle speeds with

  SPINDL/3000, CLW

- determination of the coolant with

  COOLNT/MIST

  COOLNT/OFF

To programme a drafting machine commands are available like:

PEN/... (pen number)

PENDWN (pend down)

PENDUP

DOTTED

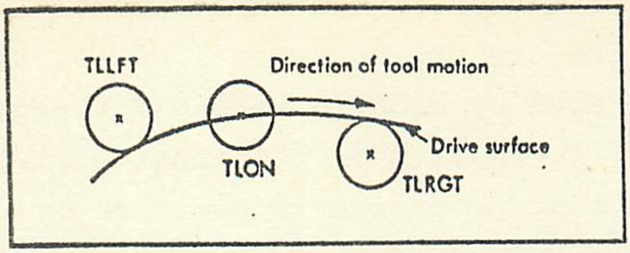


Fig. 5 Relation of tool to the drive surface during motion.

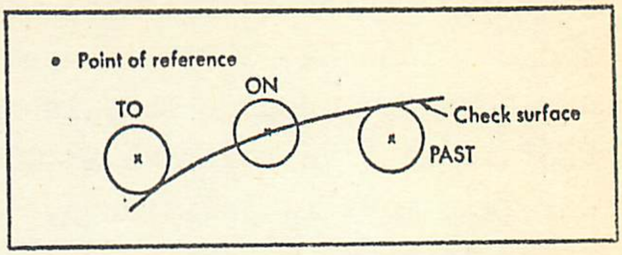


Fig. 6 *Relation of tool* to the check surface at end of motion.

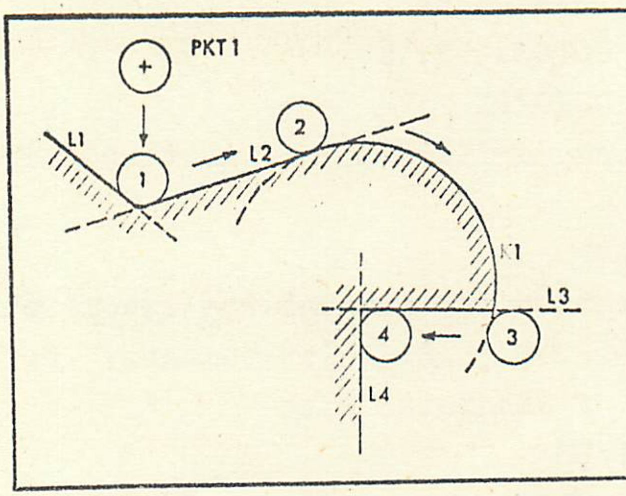


Fig. 7 Tool is guided round this workpiece by the following motion statements:

```
FROM/PKT1
GO/TO,L2,ON,SRF
GOLFT/L2,TANTO,K1
GOFWD/K1,PAST,L3
GORGT/L3,TO,L4
GOTO/P1
```

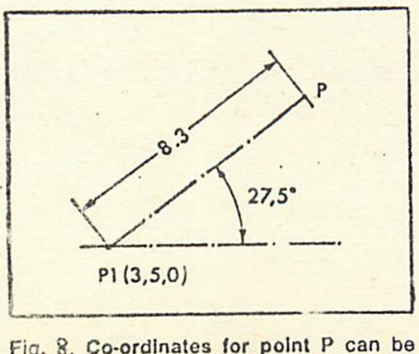


Fig. 8. Co-ordinates for point P can be calculated from following statement by formulas within the computer:

```
P1=POINT/3, 5, 0
X =3+8.3* COSF (27.5)
Y =5+8.3* SINF (27.5)
P =POINT/X, Y, O
```

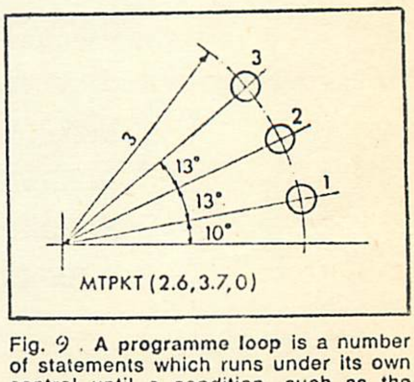


Fig. 9. A programme loop is a number of statements which runs under its own control until a condition, such as the defining of a given number of hole centres (below), is fulfilled.

```
    LOOPST
    ALFA=10
A)  X    =2.6+3 COSF (ALFA)
    Y    =3.7+3 SINF (ALFA)
    GOTO/X, Y, O
    IF (ALFA-36) B, C, C
B)  ALFA=ALFA+13
    JUMPTO/A
C)  LOOPND
```

The statement MACHIN/... is used to indicate for which machine tool a part programme is intended and with which post-processor it is to be further processed. A part programme for a Sundstrand machine tool would contain:

MACHIN/SUNTRN...

where possibly required type indicators and so on may be added.

Computer calculations

It is possible to compute values within a part programme by a mathematical formula. The calculated values need not be included in the output, they are informally available for a particular part programme. To write the formulas, a language based on FORTRAN is used. The APT language contains the arithmetic statement of the general form.

variable = arithmetic expression

This statement constitutes a command to the computer to evaluate the expression to the right of the equal sign and to store the result as value of the variable.

The application is explained in Fig. 8. With the values given the coordinates of the point P are to be determined. Since the x and y coordinates of the point are known, two general numbers, the variables x and y, are inserted in the appropriate places of the definition of the point (line 4). In lines 2 and 3 two formulas are programmed with which x and y are calculated. Line 2 says that the product of the distance (=8.3) and the COS 27.5 deg (COSF (27.5)) is to be added to line 3 and that the result is to be stored for x. The meaning of line 3 is similar.

Both formulas are evaluated during the processing of the point P is defined with the coordinates 10.36, 5.83,0.

Besides the already mentioned operators + and * for addition and multiplication there are the operators -, /, ∏, ** for subtraction, division, and exponentiation, respectively. In addition to the functions sin a and cos a the functions arc tg b, $\sqrt{x}$, exp x, $\log_e$ x with the calls ATANF (B), SQRTF (X), EXPF (X), LOGF (X), respectively, are available.

Programme logic

In a part programme often a number of statements is needed several times. A part programme may be shortened considerably, if these statements do not have to be programmed in the required number, but if they, after being coded once, are generated automatically within a part programme.

Here two methods will be described with which this aim is achieved. Both methods will be explained using point-to-point part programmes since these are most obvious. But the facility is not restricted to point-to-point work.

The first method is the formation of a programme loop. A programme loop, or simply a loop, is a part of a programme, consisting of a number of statements, which runs under its own control until a numeric condition, for example, the number of passes, is fulfilled.

Coming from an undefined point, a tool, that is the end point of the tool axis, is to go to the points numbered 1 ... 3 on a circle in the xy plane Fig. 9. The cricle is defined by its centre MTPKT (2.6, 3.7, 0) and its radius of 3. Between two successive points and the centre an angle of 13deg. is enclosed. To simplify the programme it is assumed that the tool only moves to these points. Drilling motions were not included since they do not help in the understanding of the programme loop. The most important statement (line 5) is GOTO/X, Y, 0. Because of this statement the tool centre is moved to the point the coordinates of which are given after the oblique line. Since only the z coordinate (=0) is known, again two variables X and Y are inserted.

The variable ALFA will always represent the angle between the line through a point and the centre and the horizontal X axis. In line 2 of the loop it is set to the starting value of "=10deg., the angle of point 1. In line 3 and 4, the values for X and Y are computed as has been explained before. In line 5 these values are used to move the tool to the appropriate position.

In line 6 a test is made. Inside the parentheses of the statement the number 36 is subtracted from ALFA (in the first pass for point 1 it is 10, in the second pass for point 2 it is 23, in the third pass, for point 3, it is 36). After the parentheses three labels are given which, depending on the sign of the result of the expression inside the parentheses, indicate what action has to be taken.

In the first pass (ALFA=10) the sign of the result in the IF-statement is negative (10-36 = -26). The first of the three labels is used to continue. It points to the statement in line 7. The angle ALFA is increased by 13 and the programme continues because of the statement JUMPTO/A, a so-called unconditional branch, with line 2.

During the second pass the coordinates for point 2 are determined. Again the result of the expression in parentheses in line 6 is negative, the angle is increased once more and the programme continues at line 2 again with the third pass.

Now the result in the IF-statement becomes zero. Therefore the second label is used to continue. This points to line 9.

The third label is used when a positive result is obtained in the IF-statement. In the example, the result cannot become positive. Since a label is required for the format, one of the labels used was inserted.

The statements LOOPST (=loop start) in line 1 and LOOPND (=loop end) in line 9 are auxiliary statements, required by the APT processor, to recognize the range of a loop.

The second method to eliminate time consuming coding of repetitively occurring programme steps is the use of subroutines.

Using subroutines

A subroutine is a programme, in itself complete, which cannot start its operation by itself, but which has to be called by a different programme. It is provided with input parameters by the calling programme. It processes the input parameters and at the end of its operation returns results to the calling programme and gives control over the computer back in the calling programme.

The example in Fig. 10 explains the use of subroutines within the APT system. The points are to be reached in the sequence of their numbering, however, not one after the other but from different locations in the part programme.

In line 1 ... 5 of this figure, the subroutine is defined. The word MACRO is an APT word and appears in the first statement of the subroutine to be defined. MOV is the name for this subroutine. Later on it is referenced with this name.

Following the oblique line, those variables of the subroutine are listed, which are to receive values from the calling programme. Line 5 contains the word TERMAC (=termination macro) which indicates to the processor the end of the subroutine.

A subroutine may consist of almost any number of statements and it is not restricted to a few statements. The subroutine is here followed by the main programme. At a certain location in this programme, at line 10 the subroutine MOV is called by CALL/MOV... In this call the variables of the previously defined subroutine appear with the values for point 1. After a number of statements MOVis called again (line 20), this time with the values for point 2. Similarly the values for point 3 are given in the call of the subroutine in line 30. This is followed by the remainder of the main programme.

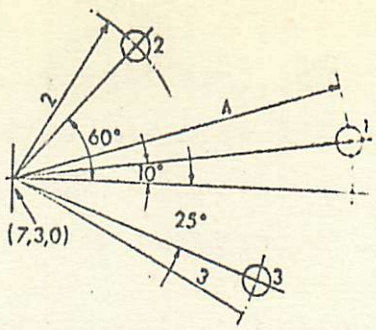


Fig. 10 Subroutine (MACRO) is here used to reduce part programming effort

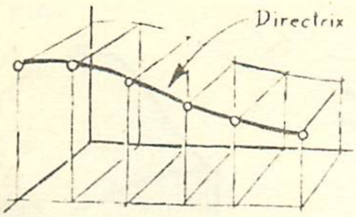
```
MOV=MACRO/RAD, ALFA
X   =7+RAD * COSF (ALFA)
Y   =3+RAD * SINF (ALFA)
GØ TØ/X, Y, 0
TERMAC
CALL/ MOV, RAD=4, ALFA=10
CALL/ MOV, RAD=2, ALFA=60
CALL/ MOV, RAD=3, ALFA=-25
```

```
PARTNO          3-D-PART PROGRAM      APT III
        NOPOST
        CLPRNT
REMARK
REMARK             GEOMETRIC DEFINITIONS
BALL     = SPHERE/ 0,0,0, 4
WALL 1   = LINE/ 0.5,0,0 , 0.25,2.625,0
WALL 2   = LINE/ 0,2,0  ,  4,1,0
WALL 3   = LINE/ 0,-4,0  ,  4,0,0
CYL      = CIRCLE/YSMALL,WALL2,YLARGE,WALL3,RADIUS,1.375
XZPLN    = LINE/0,0,0 ,  1,0,0
STPNT    = POINT/ 2,2.5,5
REMARK
REMARK            DEFINITION OF THE CUTTER, A BALL-END MILL
REMARK
           CUTTER/0.75
REMARK              TOLERANCE
           OUTTOL/0.001
           INTOL/0
REMARK              TOOL MOTION AND AUXILIARY FUNCTIONS
           SPINDL/ 3000,CLW
           FEDRAT/ 50
           FROM/ STPNT
           GO/TO,WALL1,OM,BALL,PAST,XZPLN        $$    POSITION 1
           TLRGT,GORGT/ WALL1,TO,WALL2           $$             2
           GORGT/WALL2,TANTO,CYL                 $$             3
           GOFWD/CYL,TANTO,WALL3                 $$             4
           GOFWD/ WALL3,TO,WALL1                 $$             5
           GORGT/ WALL1,ON,XZPLN                 $$             6
           GOTO/ START
           END
           FINI
```

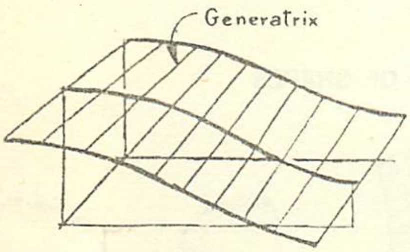
Fig. 12 … programme defines to the computer the machining of the part … 13.

Using the APT Routines

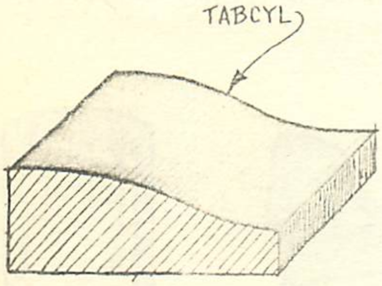
CURVE AND SURFACE FITTING



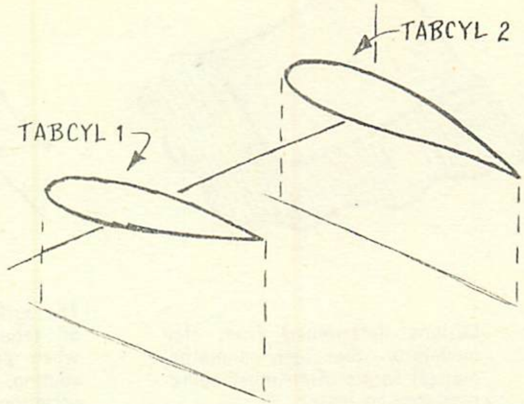
The computer provides a spline fit between points represented by ordered tabular data to generate a smooth curve called a directrix.



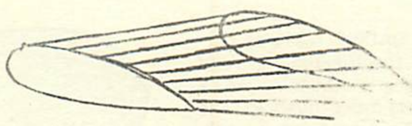
Through the TABCYL routine, a line, called a generatrix, is moved parallel to itself along the directrix to generate a surface.



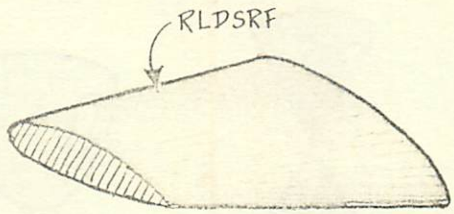
The result is a machined contour achieved by specifying only tabular da



The RLDSRF routine is used if two contours must be joined by a smooth transition surface.



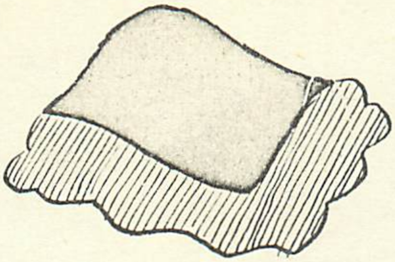
The routine generates a surface comprised of straight lines joining the two contours.



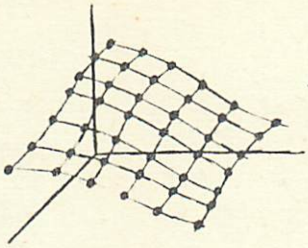
The result is a smooth transition that requires no manual computations or drafting layout.

FIG. 11

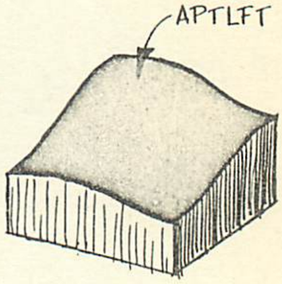
DEFINING A FREE FORM



Designs determined from clay models or other such nonmathematical forms often must be reproduced in metal.



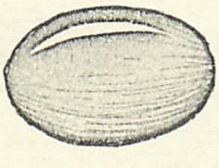
The surface is reduced to a set of tabular coordinates which, when processed by the FMILL routine, provide a numerical description of the surface.



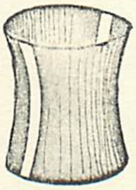
Through the APTLFT subroutine, the FMILL surface is transformed into a machined surface.

CHOOSING FROM A CATALOG OF SHAPES

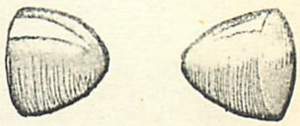
The QADRIC routines permit specification of any surface conforming to a second-order quadric. The designer need specify only the coefficients of the quadric equation. Some of the more useful shapes possible with this routine are shown here.



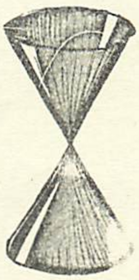
Ellipsoid



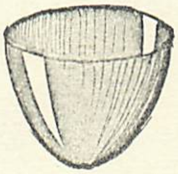
Hyperboloid of One Sheet



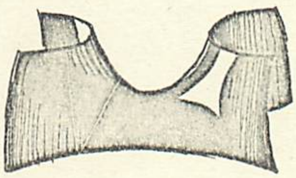
Hyperboloid of Two Sheets



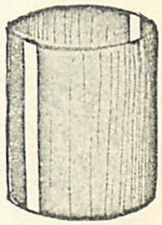
Elliptic Cone



Elliptic Paraboloid



Hyperbolic Paraboloid



Elliptic Cylinder

FIG. 11

The APT Routines

One of the major design advantages of computer programming is that complex surfaces can be specified easily and rapidly. Surfaces available through the APT routines, Fig.11, are examples.

Space Curves: Any curve in space can be represented as a series of xyz coordinates (most conveniently given in tabular form). By utilizing the APT tabulated cylinder (TABCYL) routines, the computer provides a smooth curve between these points, and calculates the intermediate points necessary to reproduce the resulting curve to any desired accuracy. The surface is generated by moving a line (generatrix) parallel to itself along the curve (directrix) which is fitted between the input points.

The TABCYL routine allows the designer to specify any desired part contour as a series of points. These points can be selected to represent the most complex mathematical curves conceivable. The computer not only produces a smooth curve between input points which the designer previously had to fit with a French curve or ducks and splines, but also produces a numerical description of that curve sufficient to allow a numerically controlled cutting tool to reproduce the curve to close tolerance. The two-dimensional applications of this feature are obvious (for cams, patterns, etc.), but there are less obvious and perhaps more important three-dimensional applications.

Surfaces Between Space Curves: Suppose a smooth transitional surface is desired between two contours of an aircraft wing. One contour (TABCYL 1, Fig.11) could be described by the designer as a tabulated collection of points. The second contour (TABCYL 2) would also be described by tabular data in the same coordinate reference system as TABCYL 1.

The designer's job is then complete. Now the part programmer describes the two TABCYL contours in the APT language. He then specifies that APT generate a ruled surface (RLDSRF) between the two contours. The computer then calculates a surface composed of straight line rulings between TABCYL 1 and TABCYL 2. This information then is used to guide a cutting tool around the desired contours. The side of the cutter is always in simultaneous contact between the two specified contours. This combination of APT features is widely used in the design and manufacture of modern aircraft components.

Quadric Surfaces: General quadric surfaces can be generated by APT. A quadric surface is defined as a surface that satisfies some particular equation of the second degree in three variables. The general form of equation for a quadric surface is $Ax^2 + By^2 + Cz^2 + Fyz = Gxz + Hxy + Px + Qy + Rz + D = 0$.

The APT feature QADRIC allows the description of surfaces by specification of the coefficients for this equation. Fig. 2 illustrates a few of the quadric surfaces which can be produced.

Quadric surfaces are inherent in the functional operations of many types of parts. For example, the QADRIC feature is used to produce molds for optical devices such as lenses and reflectors.

Free Forms: Sometimes it is necessary to reproduce surfaces which cannot be described mathematically or for which the mathematical description is extremely complex. This situation is frequently encountered where the desired contours are determined according to aesthetic considerations. (Shapes for auto-body panels, for example, are determined from clay models.) Also, complex or "Nonmathematical" surfaces are often required to blend between mathematically defined surfaces. These situations can be handled by the APT-associated FMILL and APTLFT routines, Fig.11.

FMILL is a completely independent computer routine that generates a smooth surface between an ordered set of input points. For example, assume a designer has produced an aesthetically acceptable contour in clay. It is now necessary to produce the tooling (die) required to reproduce the shape. All that is required as input to the FMILL computer program are sets of xyz coordinates taken from the surface. FMILL fits a smooth surface between this mesh of points. The output of FMILL, a numerical description of the desired surface, is then processed by APT (utilizing a special-purpose subroutine called APTLFT) to define a cutter path necessary to reproduce the surface.

3D Part programming example

In this a complete part programme and the output of the calculated cutter coordinates will be discussed.

The contour to be machined is shown in Fig.13 and the part-programme is shown in Fig.12. The last cut has to be taken along

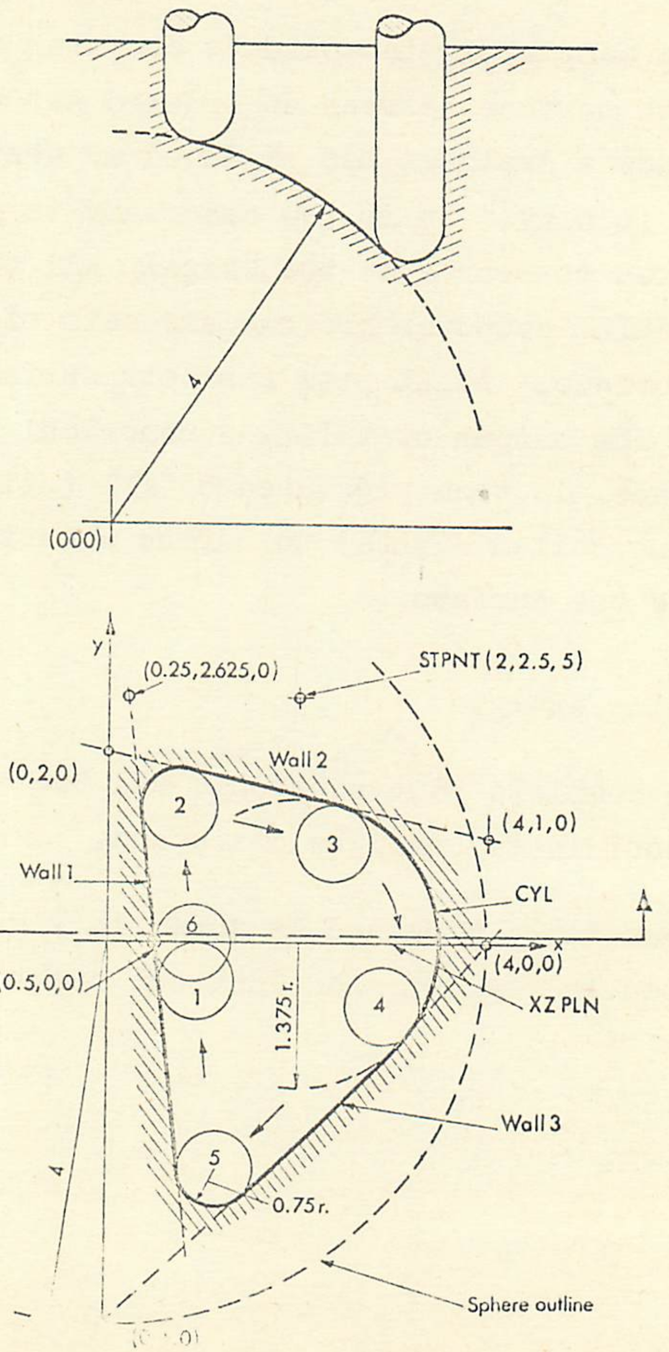


Fig. 13. Machining a pocket on a sphere provides a continuous path programming example.

the walls of the pocket on the surface of a sphere.

Although, as was said before, the statements of the APT Language have a free format, they were, in this example, punched in a certain pattern, to make it easier to read them. The part programme begins with a statement PARTNO.... This word, which has to be punched in cards columns 1...6, may be followed by any text. This statement has to be the first statement in a part programme and it is generally used to describe the part material, number of drawing, and so on.

The following statements NOPOST and CLPRNT, are indicators for the processor. NOPOST indicates that the general solution of the part programme is not to be processed by a postprocessor. CLPRNT indicates that a listing of the CLTAPE is desired.

At several places throughout the part programme there are statements beginning with REMARK. With them it is possible to insert explanatory remarks between statements to make the part programme more understandable. The word REMARK has to be punched in columns 10...60. It may be followed by any text.

The lines 60 to 120 contain the geometric definitions. A sphere (=SPHERE) is being defined by its centre coordinates and by its radius. Then the boundaries of the pocket, WALL 1, WALL 2, WALL 3, are defined as lines (=LINE). This is possible since the walls are perpendicular to the xy plane.

The points used in the definitions are indicated on the drawing. In line 100 a cylinder, perpendicular to the xy plane, is defined. It may, therefore, be defined as a circle. The

	X	Y	Z	
3-D-PART PROGRAM APT III				
CUTTER/ 0.7500				CARD NO.010
OUTTOL/0.00100000.00100000.0010000				CARD NO.160
INTOL/0. 0. 0.				CARD NO.180
SPINDL / 3000.0000 CLW				CARD NO.190
FEDRAT / 50.0000				CARD NO.210
FROM / STPNT				CARD NO.220
				CARD NO.230
	2.0000000	2.5000000	5.0000000	
DS IS/ WALL 1				CARD NO.240
	0.9124111	−0.3750000	3.8773808	
DS IS/ WALL 1				CARD NO.250
	0.8955405	−0.1978585	3.8943627	
	0.8786138	−0.0201279	3.9031826	
	0.8616664	0.1578197	3.9038219	
	0.8447340	0.3356094	3.8962793	
	0.8278521	0.5128698	3.8805705	
	0.8110560	0.6892284	3.8567284	
	0.7943810	0.8643164	3.8248030	
	0.7778621	1.0377651	3.7848614	
	0.7615337	1.2092128	3.7369871	
	0.7490980	1.3397873	3.6947291	
	0.7406691	1.4282916	3.6631399	
DS IS/ WALL 2				CARD NO.260
	0.9127025	1.3852832	3.6407648	
	1.0834753	1.3425900	3.6098112	
	1.2525965	1.3003097	3.5703502	
	1.4196595	1.2585440	3.5224761	
	1.5842726	1.2173907	3.4663012	
	1.7460476	1.1769469	3.4019579	
	1.9046013	1.1373085	3.3295986	
	2.0595627	1.0985681	3.2493927	
	2.2105652	1.0608176	3.1615299	
	2.3543358	1.0248749	3.0682158	
DS IS/ CYL				CARD NO.270
	2.4268135	1.0036083	3.0183696	
	2.5020188	0.9751207	2.9658475	
	2.5829805	0.9363042	2.9083950	
	2.6604362	0.8901666	2.8525250	
	2.7337114	0.8370078	2.7988314	
	2.7948031	0.7841255	2.7534405	
	2.8513076	0.7266353	2.7109515	
	2.9029680	0.6648186	2.6[illegible]17046	
	2.9494545	0.5990896	2.[illegible]60999	
	2.9904750	0.5298849	2.6[illegible]5144	
	3.0257746	0.4576623	2.5772947	
	3.0551381	0.3828962	2.5547480	
	3.0808968	0.2962079	2.5352636	
	3.0986905	0.2077066	2.5223407	
	3.1083987	0.1184604	2.5161582	
	3.1100785	0.0292667	2.5167007	
	3.1038628	−0.0591928	2.5238381	
	3.0899245	−0.1463134	2.5373577	
	3.0684569	−0.2315444	2.5569872	
	3.0396658	−0.3143634	2.5824077	
	3.0037710	−0.3942611	2.6132622	
	2.9610143	−0.4707231	2.6491572	
	2.9314390	−0.5158503	2.6735308	
	2.9004992	−0.5579051	2.6987125	
	2.8597508	−0.6070946	2.7313856	
	2.8180633	−0.6514707	2.7642924	
DS IS/ WALL 3				CARD NO.280
	2.7059280	−0.7637419	2.8463772	
	2.5906350	−0.8790349	2.9194380	
	2.4725984	−0.9970715	2.9831721	
	2.3522046	−1.1174653	3.0373698	
	2.2298384	−1.2398316	3.0818580	
	2.1058891	−1.3637808	3.1164953	
	1.9807537	−1.4889162	3.1411708	
	1.8548329	−1.6148371	3.1558052	
	1.7285275	−1.7411425	3.1603522	
	1.6022432	−1.8674268	3.1547970	
	1.4763824	−1.99328[illegible]	3.1391575	
	1.3513477	−2.1183223	3.1134836	
	1.2275386	−2.2421314	3.0778574	
	1.1371435	−2.3325266	3.0452304	
	1.1021728	−2.3674972	3.0310680	
DS IS/ WALL 1				CARD NO.290
	1.0887756	−2.2268271	3.1405430	
	1.0749174	−2.0813158	3.2434363	
	1.0606272	−1.9312690	3.3395316	
	1.0459349	−1.7769999	3.4286284	
	1.0308715	−1.6188335	3.5105390	
	1.0154684	−1.4571009	3.5850917	
	0.9997578	−1.2921403	3.6521308	
	0.9837729	−1.1242985	3.7115153	
	0.9675471	−0.9539276	3.7631206	
	[illegible]9511143	−0.7813828	3.8068391	
	[illegible].9345089	−0.6070265	3.8425790	
	0.9177059	−0.43122[illegible]	3.[illegible]702061	
	0.9009204	−0.2543478	3.8898395	
	0.8890963	−0.1301947	3.8986867	
	0.8766968	−0.	3.9036654	
DS IS/ STPNT				CARD NO.300
	2.0000000	2.5000000	5.0000000	
END				CARD NO.310
FINI				CARD NO.320

Fig. 14 This is the printout of the CITAPE from the computer. It is the data for controlling the path of the cutter. It now needs post-processing to fit into a particular system/machine tool combination.

circle is tangent to WALL 2 on the side of the smaller y coordinates (YSMALL), whereas it touches WALL 3 on the side of the larger y coordinates (YLARGE). By giving the radius, the circle or rather the cylinder is uniquely defined.

The tool to be used is a ball end mill with radius 0.75in. (line 160). In line 180 the outside tolerance is given by OUTTOL/ 0.001. An undercutting of the programmed contour is prevented by INTOL/0.

The description of the tool motion begins with line 230, after the spindle speed and the feedrate have been programmed in the two previous lines. The startup (line 240) brings the tool into position 1, that is close to WALL 1, so that the tool axis end point lies on the sphere and behind the xz-plane (XZPLN). The remainder of the statements is easily understood with reference to the earlier section on tool motion statements. The statement END (line 310) is a command for the post-processor, which stops various functions such as the spindle rotation and the coolant flow. The last statement of a part programme must be FINI. The card resulting from it becomes the last card of the part programme. It indicates to the processor the end of the part programme and it starts the execution of further processing passes.

The result

The result of the processing of the part programme, that is the general solution, is shown in the computer printout in Fig.14. As well as being printed out it is coded for further processing on the CLTAPE.

The printout of the general solution consists of a reproduction of input cards and of calculated coordinate values. On the right side of the list the number of the part programme card from which values were taken or whose contents resulted in the shown coordinates, is printed.

The list starts with a reproduction of the text of the PARTNO card and with a repetition of the tool description, of the tolerance statements and of the programmed machine functions. Here the tolerance statements contain three values, although only one value was programmed. It is possible to programme different tolerance values for each of the three guiding surfaces. If, as in the example, only one value was programmed it will be used for all three surfaces. After the FROM statement a line containing the three coordinates of the starting position follows.

From card 240, Fig.14, which contains the startup, position 1 (to the right of WALL 1, tool centre behind the xz plane, and the end of the tool axis on the sphere) is calculated and WALL 1 is assigned to be the next drive surface: DSIS/WALL 1 (DSIS = Drive surface is). Under DSIS/WALL 1, comes coordinates of the tool path along WALL 1 to WALL 2, card 250 Fig.14. From the coordinates a motion in three axis was calculated. From card 260, the coordinates of the tool path along WALL 2 were calculated.

Next come the coordinates of the tool path along the cylinder CYL, along WALL 3 and along WALL 1,as programmed in cards 270 to 290. The last coordinates are those of the point STPNT. to which the tool is moved due to statement 300. The list is terminated by the reproduction of the statements END and FINI.

The part programme contains eight motion commands. The APT system computed 80 coordinate triplets from them. Depending on the experience of the part programmer, it takes 15 to 30min. to compile this part programme. The calculations, including the output, were performed by an IBM 7094 in 10sec.