

Genesis Evolution

Control Editor

Index

Chapter I	Using manual	1
Introduction		1
Work graphic environment		1
Structure of the basic Graphic Page		2
Operating Functions		3
Program Archive		3
Access mode		3
Creating a new program		4
Working Environment		4
Panel Visualization area		6
Axes reference systems		6
General piece view		8
Face View		9
3D View		9
Fictitious Faces		10
Structure of the working data table		12
Structure of ISO Text Table		13
Main Tool Bar		14
File Handling Buttons		14
General Characteristics Setting Buttons		15
Current Program Handling Buttons		15
Zoom Buttons		15
Technology Help Buttons		16
Graphic Simulation Buttons		16
Block / Faces Selection Buttons		16
Sliding Text Buttons		17
CAD Tools Predisposition Buttons		17
Macros-Workings		17
Working Classification		18
Single and Multiple Holes		18
Simple Rotary Cuts		19
Complex Rotary Cuts		19
Saw		20
Measures and Miscellaneous		20
Subroutines Calls		20
Special Functions		20
Palettes for Working Selection		21
Level-type Structure of the Working Palettes		21
Data Entry Dialogue Box of Working Parameters		22
Position Area		23
Tool Area		24
Feed Area		25
Outfits and working Tools		26
Working Tools programming		26
Programming by Diameters and typology		26
Drilling Tools		26
Milling tools		28
Sawing Tools		30

Help on Line for Tools	31
Drawing Manipulation	32
CAD manipulation tools Palette	33
"Move" command	33
"Rotate" command.....	33
"X Symmetry" command.....	34
"Y Symmetry" command.....	34
"X/Y Symmetry" command.....	35
"Repeat" Command.....	35
"Explode" command.....	36
"Profile cutting".....	36
"Profile reversing".....	36
"Move profile Setup" command.....	36
"Profile Scaling" command.....	36
"Profile Link" command.....	37
"Profiles connection" command.....	37
"Round" command.....	37
"Beveling" command.....	37
"Entry path" command.....	38
"Exit path" command.....	38
"Z Laying".....	38
Parametric Programming	38
Variables	39
Variables assigning.....	40
Program basic Variables	41
Mathematical Expressions and Operators	41
Basic Arithmetical Operators.....	42
Mathematical transcendent Operators.....	42
Trigonometric Functions.....	43
Special and Technological Functions.....	44
Parametric Subroutines	44
Subroutines recalling functions	45
Recalling Options: Translating, Rotating and Mirroring.....	46
Subroutine Variables re-assigning	47
Chapter II Workings Programming	49
Drilling workings	49
Piece Definition and dimensions	49
DRILL (X,Y,Z) working	49
Drill Execution with the option "Relative"	51
DRILL in "Polar coordinates"	52
"Multi-tools Drilling"	53
"Repeat X", "Repeat Y", "Repeat XY" macros	53
"Fitting X" and "Fitting Y" Macros	55
"Drills on Circle" macro	55
"Drills on vertices of an inscribed Polygon" Macro	56
"Drills on vertices of a circumscribed Polygon" Macro	57
Sawing instructions	58
"Saw X", "Saw Y" and "Saw in degrees"	58
Mill Setup and Linear / Circular interpolations	60
"Mill Setup (x,y,z)" working	60
"Mill Setup" (polar coordinates)	62
"Oriented Mill" Setup	63
Linear milling - "L01 (X,Y,Z)"	64

Linear milling - "L02 (pole,A,U)"	65
Linear milling - "L03 (A,U)"	66
Linear milling - "L04 (pole,A,X)"	67
Linear millings - "L06 (A,X)" and "L07 (A,Y)"	67
Linear milling - "L08 (Z,tangent)"	68
Linear milling - "L10 (closure)"	69
Linear milling - "L12 (Tginiz (P1,P2 o A), U, Z)"	69
Arc "A01 (center) on plane XY"	70
Arc "A04 (three points) plane XY"	71
"A11 (radius, smaller arc) on plane XY"	72
Arc "A13 (center and final angle) on plane XY"	73
Arc "A15 (entry tangent) on plane XY"	74
Arc "A16 (initial tangent, final point) on plane XY"	75
Arc "A17 (exit tangent) on plane XY"	76
Double Arc "Arc1-Arc2 (C1,C2)"	76
Double Arc "Arc1-Arc2 (C1,R2)"	78
Double Arc "Arc1-Arc2 (R1,C2)"	79
Double Arc "Arc1-Arc2 (C1, exiting tangent)"	80
Double Arc "Arc1-Arc2 (Entry tangent, C2)"	82
Double Arc "Arc1-Arc2 (Entry tangent, exit tangent)"	83
Programming on sloped planes ("Oriented Mill Setup")	84
Macro	87
"Circle (center)" macro	87
"Circle (three points)" macro	87
"Oval" Macro	88
"Ellipsis X" (Ellipsis Y) macro	91
"Polygon inscribed in a Circle" macro	92
"Polygon circumscribed to a Circle" macro	93
"Fillet" macro	94
"Beveling" macro	95
Subroutines	96
Subroutine "sub1.cnc" programming	96
Subroutine "sub2.cnc" programming	98
Subroutine recall with the "SUB0" instruction	99
Subroutine recalling with the "SUB2" function	103
Appendix 1: Data Entry tables for the functions: SUB0 and SUB2	104
Appendix 2: Toolbar and Palette	106
Surface Clearing and Develop Text Tools	107
Surface Clearing Tool	107
Setting Window for Surface Clearing Parameters.....	107
Develop Text Tool	110
Setting Window for Text Parameters.....	110
Geometry of text distribution.....	111

1 Using manual

1.1 Introduction

Edicad32 is the software environment for programming "Genesis Evolution" Numeric Control, suited for driving Wood Working Machines, specifically for multifunction work centers used for customization of panels in the manufacturing of modular furniture.

Edicad32 was conceived as to offer user a very friendly and easy working environment, really "problem oriented": it proposes an assisted data introduction, easy design of working paths and integrates within the system software the main construction rules specifically related to modular furniture assembling.

Edicad32 offers a number of Programming Tools, both in the technologic environment – thanks to a rich library of Macro-Instructions suitable for the specific working cycles required by the Application – and in the graphic environment – by means of various geometric tools and drawing manipulation functions, allowing easy and safe programming of complex working paths.

Edicad32 is a 32 bit application, using COM technology in accomplishment of WindowsNT® standard requirements; it runs under WindowsNT® operating system and makes use of its MMI graphic operating modes.

Consequently, it offers a simple programming environment, based on graphic windows for visualization and data entering, equipped with Text Menus, Tool bar, Help-on-line and Dialog box for guided introduction of geometric and technologic parameters, allowing continuous and immediate check of the data introduced.

1.1.1 Work graphic environment

By exploiting the system graphic performances, and specifically use of high resolution display, most of the programming environment could be concentrated within a single screen page, based on a multi-window structure offering text menus, Tool bar and Tool Palette, thus offering an immediate and easy selection of the required commands and functions.

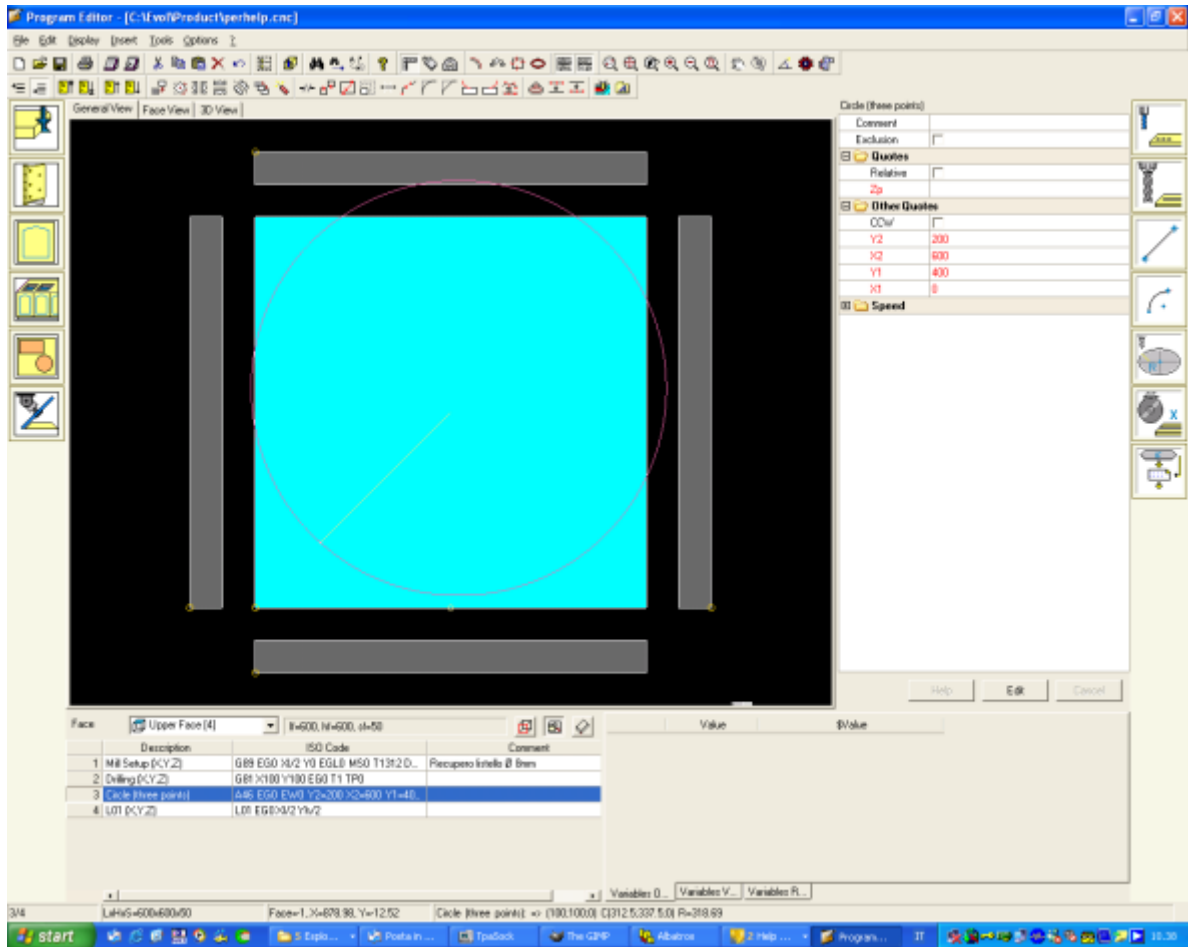


Fig. 1.1 - Work Environment

1.1.1.1 Structure of the basic Graphic Page

The main page is subdivided into the following main areas, as shown in [Fig. 1.1](#):

- Panel graphic window**: (displaying all the 5 faces that may be worked), with scaled representation (details may be Zoomed) of panel, workings inserted and routing paths.
- Dialogue Box**: for introduction of geometric and technologic parameters, guided by a structured list of options available for each type of Working or Macro-Instruction.
- Data Table for ISO Text**: displays in "ISO text " format, on a scrollable list, the program lines as soon as they are introduced
- Variables Data Table**: the list and the current value of the program Variables, used in parametric programming, are permanently displayed in this list
- Text Menus**: provides selection of general commands and functions, as File Save, Edit program, View mode, activating the corresponding palettes
- Tool bar**: organized in several levels, provides immediate selection of frequently used commands and tools, as well as recalling technologic information imported from other Applications
- Working Palette**: (on the right side) for immediate selection of macro-workings, with the

consequent display of the Dialogue Box for parameter setting

8. **Drawing Palette:** (on the right side) for immediate selection of graphic objects and geometric manipulation of profiles
9. **Status Bar:** displays panel and face dimensions and the main data referred to current working

1.1.2 Operating Functions

System performances and main operating functions are described hereafter.

Edicad32 software allows creating, editing and saving programs or subroutines, checking and optimizing them in order to provide their translation into an executable program, according to the tooling and parameters of the machine.

The Operator may freely define an archive structure for main programs and subroutines, and organize them according to his lines of furniture. Programming may be executed locally (on the Numerical Control) or in the Office on a remote workstation, and also by means of external CAD/CAM programs, saving, importing, and converting them automatically into .DXF files or ISO text.

Use of tools as Select, Drag and Drop, Recall of parametric subroutines, etc. allows easy treatment of programs and almost automatic creation of special versions or differently sized panels starting from standard models.

Parametric programming – providing introduction of geometric rules in the disposition and layout of workings- and the availability of conditioned programming (IF/ENDIF blocks) allow introduction of furniture building rules, increasing reliability in program development even in case of special parts, and reducing response time in the treatment of orders.

Edicad32 graphics environment is highly configurable; many programmable settings allow introduction of default information, possibility of enabling or masking the library of Workings and Macros, definition of rules for importing and converting .DXF files; system may be configured in the most suitable mode for the different machines and requirements.

1.1.3 Program Archive

1.1.3.1 Access mode

Easy and immediate access to program archive is provided by fruition of the native display and selection performances of WindowsNT resource manager.

The "tree" organization of the archive may be exploited, easily reaching the program or subdirectory required, thanks also to the graphic Preview that provides immediate check of what is searched (Fig. 1.2).

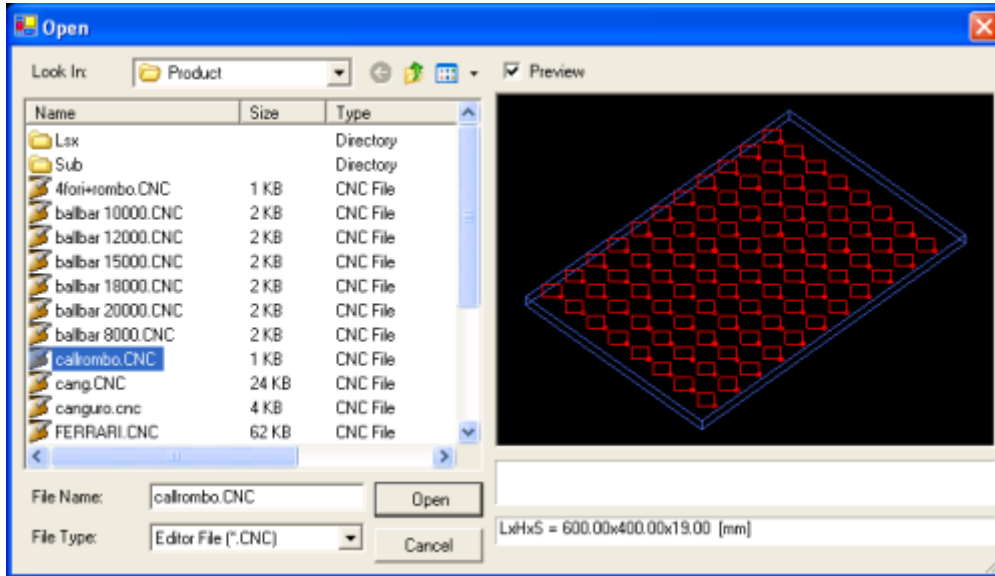


Fig. 1.2 - Selecting a program with Preview

Preview function is available, and specially useful, also when calling Subroutines, to be inserted within a main program, or when searching external programs, as in the case of .DXF or .DWG files

1.1.3.2 Creating a new program

Before creating a new program, various initial information must be input by the operator

- Programming mode and measure units ([mm] o [inch])
- Nominal piece dimensions, treated as a parallelepiped (LxHxS)
- Special shapes, as inclined faces, named *Fictitious Faces* and numbered from 7 onwards (see [Section 1.2.1.4](#))
- In this last case, each single fictitious face will be characterized by 3 main points, able to identify position, orientation and "Local" reference system adopted in programming. These information are strictly required by the system, for converting quotes of workings programmed on such faces into the equivalent "physical" quotes of machine axes.
- When saving, definition of **Name** and destination **Directory**, if different from default.

1.2 Working Environment

The working environment of EdigrafF32 provides programming of the Part program of the Numerical control "Genesis Evolution", allowing to set up the characteristics of the piece, to introduce the workings to be executed on it and to manage the archive of the job programs.

In this environment all the geometric and technological information required to characterize the job cycle may be inserted, exploiting direct access to the Parametric files of the Machine ("Outfit parameters") and using a series of functions and instruments of design of complex workings (as beveling, connections, ellipses, arcs, ovals).

Each of these functions has the scope to simplify and fasten the construction of geometric profiles on the piece, allowing to display and check the result of the programming work.

The Edicad32 environment is organized in a graphic screen where, in a structured format, various workspaces are available, oriented to graphic and text visualization and to data setting:

- the view of the panel under construction, that may be selected in General View mode or in Face View general or Sight for face, (**fig. 2.1a and fig. 2.1b**)
- the Working Parameter Setting Table, (**fig. 2,2**)
- the visualization area of the corresponding Text in ISO format (**fig. 2,3**)
- the Table for programming/displaying parametric Variables (**fig. the 2,4**)

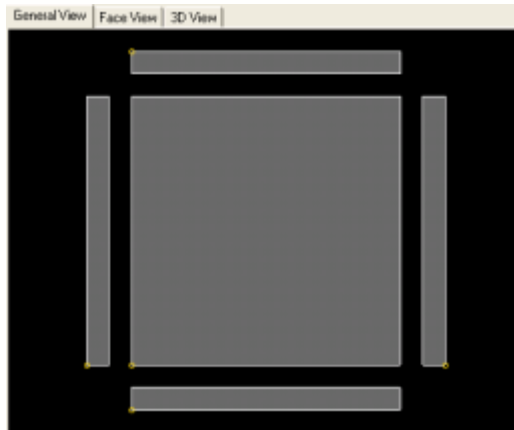


Fig. 2.1a: General Piece View
Panel is represented with the explosion of the 5 faces that may be worked: each face is associated to a local Cartesian Reference System [x, y ,z].

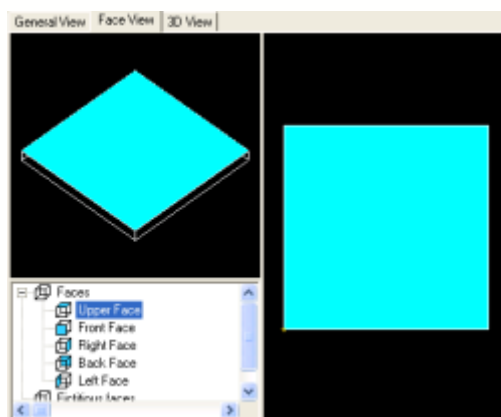


Fig. 2.1b: Face View
Panel is displayed in 3D view, with the selected face displayed in foreground, for the insertion of workings.

Fig. 2.2 shows an example of structure of the programming Table of working specific parameters,

Fillet	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
Relative	<input type="checkbox"/>
X Corner	
Y Corner	
Zp	
<input type="checkbox"/> Other Quotes	
X4	
Y4	
Arc rot. direction	Automatic
Final Angle	
Fillet width	

Fig. 2.2: Working Parameters Table

The Chart is based on a tree structure, characterised by the different typologies of parameters (geometric, technological and options) typical of the working, that allow a guided and easy programming.

Fig. 2.3 show the format of the panel of Visualization of the correspondent ISO text, that checking and editing the numerical values. For a more detailed description refer to [Section 1.7](#).

	Iso Code	Description
1	G81 X500 Y250 Z-10	
2	A15 X100 Y100 Z-10 EG1 EW2=2	

Fig. 2.3: "ISO" Text Table

In this table all the corresponding ISO lines of the program are listed: it is also possible to modify data without entering again in the Working Data Table.

Fig. 2.4 shows an example of the area of data display / setting for the use of Parametric Programming, described in detail in [Section 1.8](#).

	Value	Value
Offset X	0	
Offset Y	0	
Offset Z	0	

O. Variables V. Variables R. Variables

Fig. 2.4: Variable Table

In this table the program Variables are displayed, that can be used as local Parameters, as to offer a great facility and flexibility in the programming, with use of mathematical calculation formulas.

1.2.1 Panel Visualization area

1.2.1.1 Axes reference systems

In this View mode all the workable faces of the piece are represented contemporarily, allowing to select, one after the other, the one on which to operate.

For each face the origin of the local Reference System is enhanced, that will facilitate the definition of the quotes of application of the different workings. In fact every face will exploit a local reference system, in which the same face is prepared on the canonical Cartesian plane [XY], while the Z axis will always define the direction perpendicular to it, with the convention of sign Z+ in the direction exiting from the piece. Insofar all the indications of Depth will be characterized by negative values.

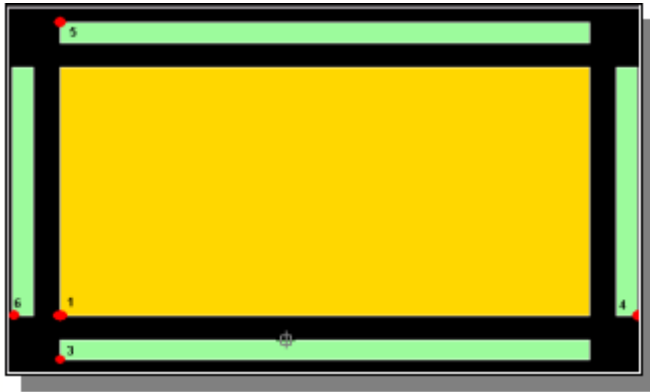


Fig. 2.5: General view

Faces are numbered according to their disposition, and on each one is enhanced the vertex corresponding to the origin of the local Cartesian reference System.

From the conventions described above it comes out that only the main face (Face 1) will introduce a system of local axes coincident with the physical machine axes [X, Y, Z]. For the other faces a different correlation will exist instead, listed hereafter, where the axes X' and Y' of local programming are identified with the terms "abscissa" and "ordinate" and with "thickness" the direction of penetration into the same face.

In synthesis, we have:

- Face 1 : abscissa \Rightarrow X, ordinate \Rightarrow Y, thickness \Rightarrow Z (Fig. 2.6)
- Face 3 : abscissa \Rightarrow X, ordinate \Rightarrow Z, thickness \Rightarrow Y (Fig. 2.7)
- Face 4 : abscissa \Rightarrow Y, ordinate \Rightarrow Z, thickness \Rightarrow X (Fig. 2.8)
- Face 5 : abscissa \Rightarrow X, ordinate \Rightarrow Z, thickness \Rightarrow Y (Fig. 2.7)
- Face 6 : abscissa \Rightarrow Y, ordinate \Rightarrow Z, thickness \Rightarrow X (Fig. 2.8)

Notice that faces 3 and 4 introduce a canonical Cartesian plane as it is seen "from the outside" of the piece. For faces 5 and 6 (for homogeneity with the preceding ones) the plane is seen "in transparency", that is as it would be visible from the inside. This allows to maintain the usual position, bottom left, of the origin of the Cartesian tern.

The following figures illustrate the disposition of the local Reference Axes.

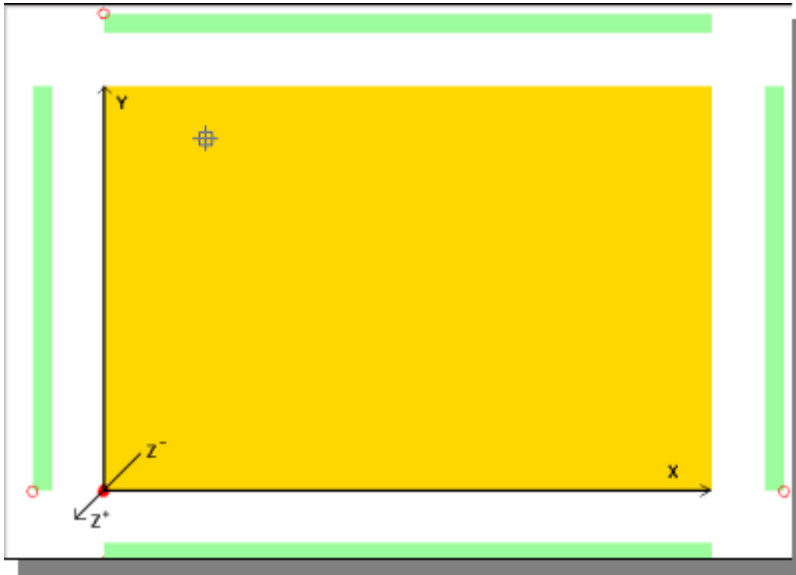


Fig. 2.6: Face 1 View

Axes x , y and z are shown in this figure. Z is positive if exiting from piece, and negative entering into material.

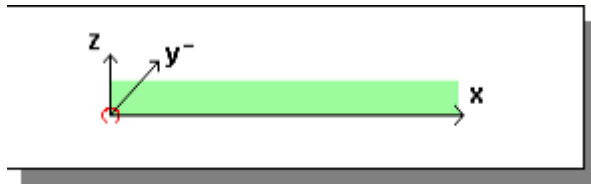


Fig. 2.7: View of faces 3 and 5

Ordinates are represented in this face by thickness and depth by piece height.

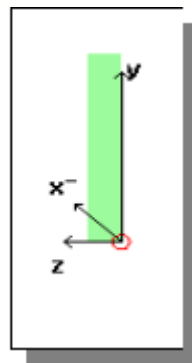
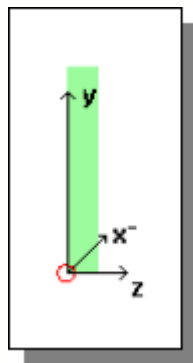
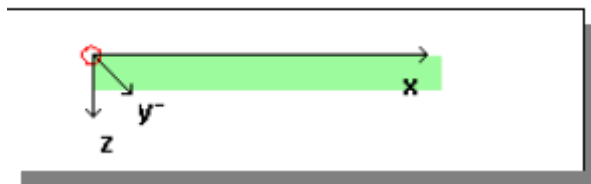


Fig. 2.8: View of faces 6 and 4

In this face abscissas are represented by thickness and depth by piece width

1.2.1.2 General piece view

The face views initially presents a 3D perspective display of workable faces, for their selection. (Fig. 2.9a-b).

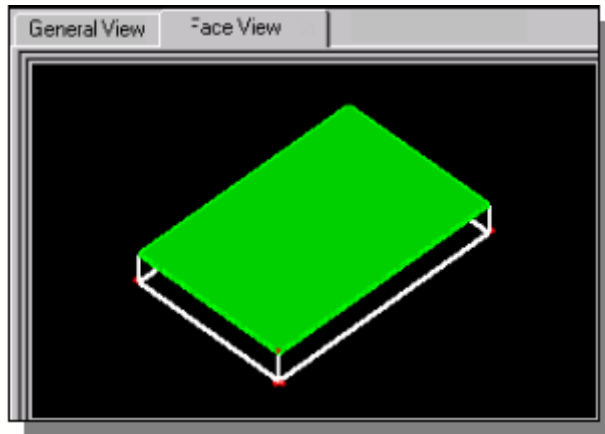


Fig. 2.9a : Piece 3D view

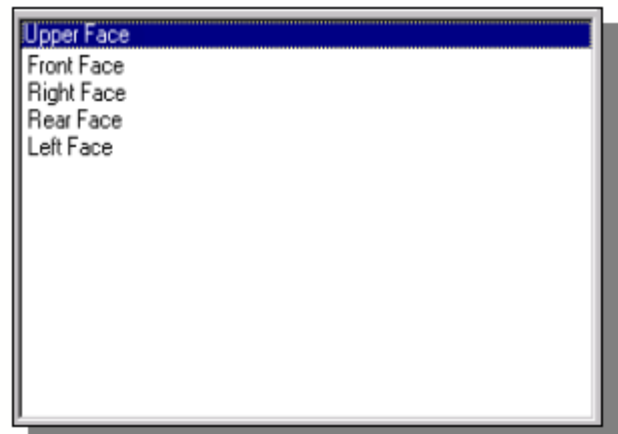


Fig. 2.9b : List of faces that may be selected

1.2.1.3 Face View

Once selected, a face is displayed in its local representation, allowing direct geometric programming: on this face the various workings introduces will be displayed (Fig. 2.9c).

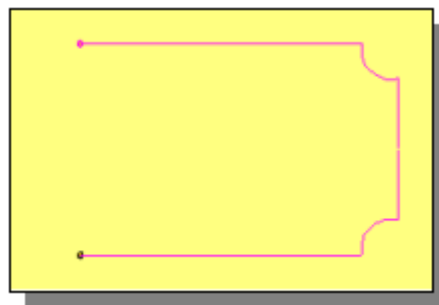
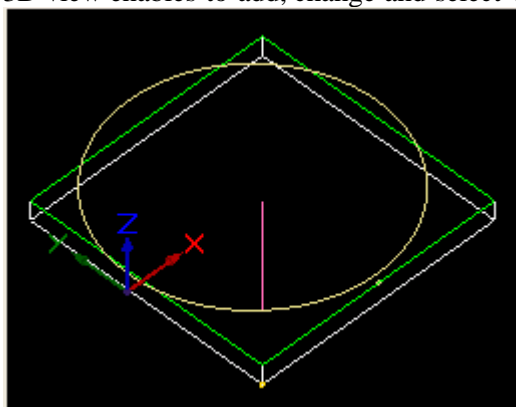


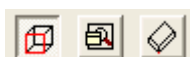
Fig. 2.9c : Face main view

1.2.1.4 3D View

3D view enables to add, change and select workings.



Three buttons are available:



their functions are:

- alignment with the perpendicular view of the selected face
- viewing the workings of the selected face only
- resetting the rotation angle of the part to 45°

This view does not allow selection by area.

1.2.1.5 Fictitious Faces

Edicad32 program also offers the possibility, for pieces of complex form, to define other work planes, beside the standard perimeter faces: these other planes, named "Fictitious Faces", can represent both inside or perimeter faces (but not coincident with the external), translated or however oriented in space, that implicates the availability of tools inclined for their working.

These additional faces must be expressly programmed, defining their position and orientation by means of an orderly sequence of 3 characteristic points.

The formulation of an additional face is achieved, as shown in Fig. 2.10, through Menu "Edit" with the subsequent definition of the 3 notable points (X_0, Y_0, Z_0) , (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) defining it.

Every fictitious Face introduced will assume a progressive number beginning from 7.

We will see later the logic with which these three points must be assigned, and their sequence.

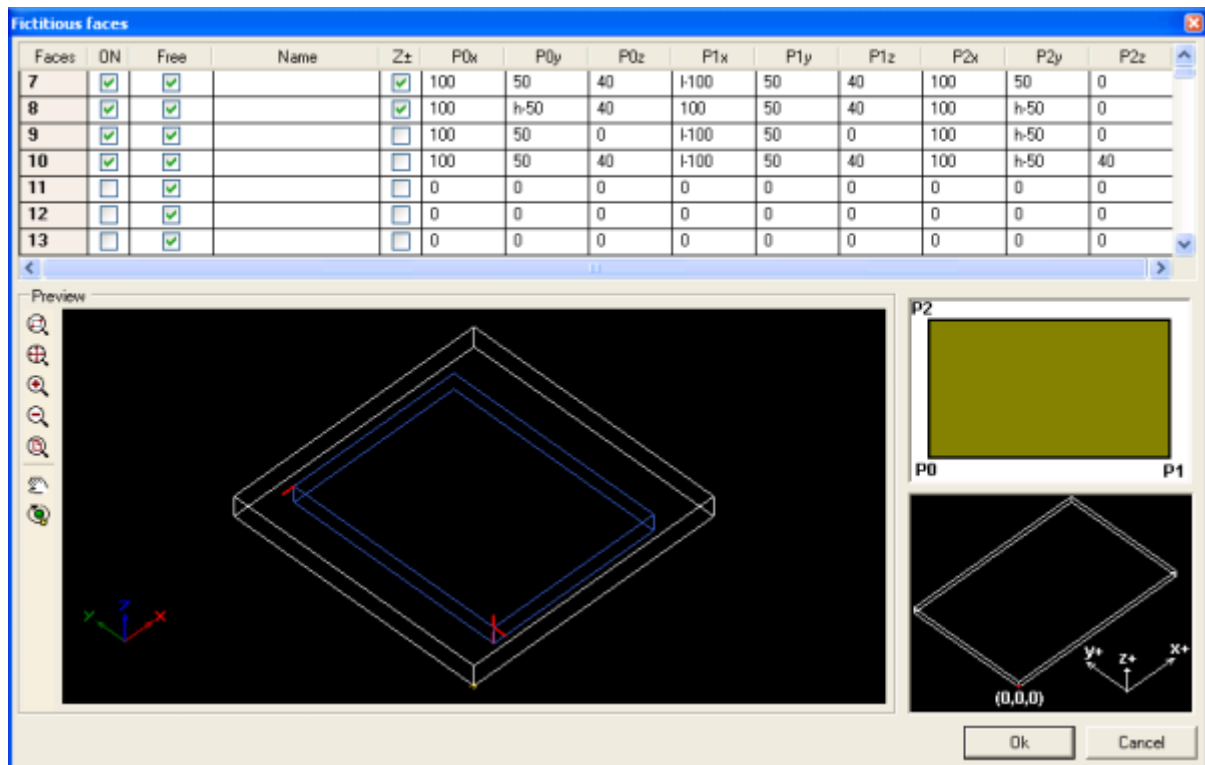


Fig. 2.10: This figure shows the Table for setting points P0, P1, P2 that will define the fictitious face, the 3D visualisation of the panel and the specific representation of the current fictitious face.

The 3D view of the piece may be modified, rotating it as illustrated in Fig. 2.11.

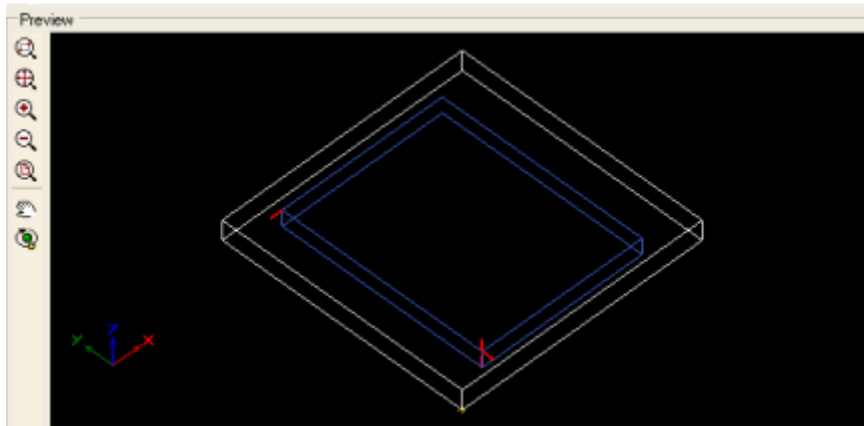


Fig. 2.11: from this view it is possible to see in detail how the inside faces of the piece have been positioned. The reference origin always remains the same and, for the fictitious faces, it is possible to define the direction of the Z axis. Inside this piece 6 internal faces were built. There is also a small option bar: with command 3D rotation it is possible to rotate the piece in any direction.

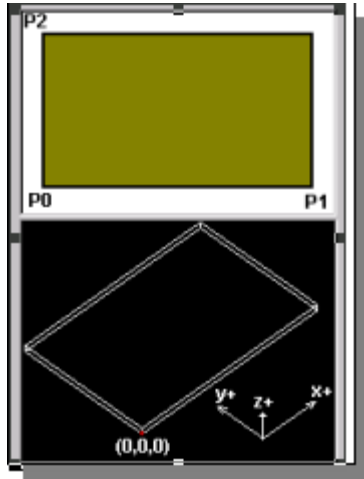


Definition of the 3 notable points of the fictitious face is very important, as it determines the position and orientation of the face but also, implicitly, the choice of the local reference system.

With reference to Fig. 2.12, the following rules must be observed when programming points P0, P1 and P2:

- First, the 3 points must not be lined up, because in such case the definition of the plane is not univocal
- Their sequence of introduction must be assigned in such way that the circuit $P0 \rightarrow P1 \rightarrow P2$ is counter-clockwise, that implicitly provides a view "from the air", that means with Z+ exiting toward the operator.
- Point P0 must be programmed considering that it will be chosen as Origin of the local axes
- Point P1 will be assumed by the system as indication (from the oriented segment $P0 \rightarrow P1$) of the direction of local axis **X+**
- The third point P2 (counter-clockwise circuit) will allow the system to deduce the **exiting Normal Z+**, therefore the real orientation of the plane
- The system will automatically calculate the orientation of axis Y+, perpendicular to the other two axes.

The reference system so built is what the Operator must consider when programming quotes of the Workings on the mentioned face.

**Fig. 2.12:**

The first figure shows a correct disposition of the 3 points of definition of the fictitious face and a representation 3D of the piece, as to facilitate comprehension of the mechanism of quoting its points (in absolute reference)

Fig. 2.13 illustrates an example of fictitious faces inside the piece, oriented in the directions of the Cartesian axes.

Face	ON	Libera	Nome	Z±	X0	Y0	Z0	X1	Y1	Z1	X2	Y2	Z2
7	X	X		X	100	50	40	1-100	50	40	100	50	0
8	X	X		X	100	h-50	40	100	50	40	100	h-50	0.0
9	X	X			100	50	0	1-100	50	0	100	h-50	0.0
10	X	X			100	50	40	1-100	50	40	100	h-50	40

Fig. 2.13 – Inserting notable points of the fictitious face

Fields:

- [ON] : Enable fictitious face
- [Free] : Indicates whether Workings are present on this face
- [Z+] : Indicates Z direction considered in the workings, if present

1.2.2 Structure of the working data table

In this table are inserted all the geometric and technological parameters that characterize the various Workings, that means:

- Working Application Point (in face X, Y, Z co-ordinates), in Absolute or Relative mode
- Other characteristic geometric parameters , if any (Quotes and/or Angles)
- Diameters and/or Tools required
- Working Speed or other Technologic Parameters

- Note : Some data may be optional, other not compulsorily programmable, because assumed by system as default.

- Note : Some workings may be momentarily disabled, when non compatible with the sequence or the face being programmed (for instance, on side faces 3, 4, 5, 6, Saw workings are not allowed).

Refer to [Section 1.4](#) for an analytic description of the various fields, and to [Part 2 - Workings](#), for the corresponding technological meaning.

Some examples of programming tables are shown hereafter (Fig. 2.14a-b-c):

Drilling (X,Y,Z)		Saw oriented		Mill Setup (polar coordinates)	
Comment		Comment		Comment	
Exclusion	<input type="checkbox"/>	Exclusion	<input type="checkbox"/>	Exclusion	<input type="checkbox"/>
[-] Quotes		[-] Quotes		[-] Quotes	
X		Xi		Relative	<input type="checkbox"/>
Y		Yi		Zp	
Zp		Alfa angle		[-] Other Quotes	
Relative	<input type="checkbox"/>	Beta angle		X Centre	
[+] Other Quotes		Module		Y Centre	
[-] Tools		Zp		Angle	
Diameter		[+] Other Quotes		Module	
Tool	1	[-] Tools		[+] Other	
Tool type	Blind bore dri	Tool 1		[-] Tools	
		Tool radius compensation	None	Tool 1	
		[+] Speed		[-] Tool radius	
		[+] M Fields		Tool radius compensation	None
				Tool radius value	
				Progressive Tool Radius In	<input type="checkbox"/>
				Progressive Tool Radius Out	<input type="checkbox"/>

Fig. 2.14a
Working data table for function **Drill**, with characteristic parameters and optional fields.

Fig. 2.14b
Working data table for function **Angular Saw**.

Fig. 2.14c
Working data table for function **Mill Setup** (in polar coordinates).

1.2.3 Structure of ISO Text Table

This control Panel is conceived for visualizing the equivalent ISO text of the program under construction. The fully grown ISO foresees to characterize every Workmanship through a lace of characters, whose syntax reflects the rules of planning of the Language, opportunely you adapt for codifying the different specific operations.

The ISO line corresponding to the current Working is enhanced with a color band.

Text may be edited, in order to modify its content: double click with mouse on the line selected, then edit text.

ISO text may be printed, as to get a quick report of all workings; its comprehension is made easier by typing into field "description" a comment associated to working. Text of each ISO line contains all the information associated to working, in accordance to the usual language rules. As an instance, the description of code associated to Drilling working is shown in the following example:

	Description	ISO Code	Comment
1	Mill Setup (X,Y,Z)	G89 EGO XI/2 YO EGL0 MS0 T1312 D...	Recupero listello Ø 8mm
2	Drilling (X,Y,Z)	G81 X100 Y100 EGO T1 TP0	
3	Circle (three points)	A46 EGO EW0 Y2=200 X2=600 Y1=40...	
4	L01 (X,Y,Z)	L01 EGO XI/2 Yh/2	

- **G81**: Function G.. represents, as usually, the code that identifies working type: in this case G81 represents working DRILL
- **X100**: X application quote of drilling

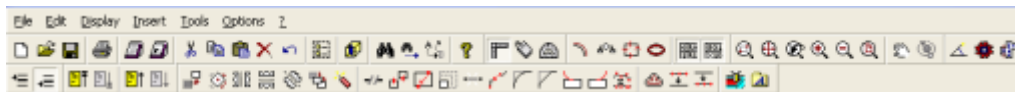
- **Y100**: Y application quote of drilling
- **Z-10**: Drilling depth
- **EG0** : "Absolute" programming mode (EG1 corresponds to "Relative")
- **A0** : Indicates that direct tool selection was applied (and not programming by Diameter =A1). In case of A1 the following fields TR.. and T.. would be disabled
- **TR1** : Group Number (=1)
- **T1** : Tool Number (=1)

1.3 Main Tool Bar

Edicad32 programming environment has the typical WindowsNT software structure used for Command Data Entry.

Commands may be selected either using written "pop-up" Menus or, more quickly, by directly using the working Tool Bars, which are made both in a fixed structure, just as the Main Tool Bar described in this chapter, and in command Palettes which can be personally set and are sensitive to the context in which they are located.

Edicad32 Main Tool Bar gives the user not only a quick and easy access to the most frequently used commands but also helps in different functions.



Let's now see a punctual description of the different command buttons, on the basis of their actions.

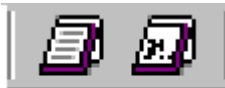
1.3.1 File Handling Buttons



They let you handle Directories and Files that constitute the work Part Programs:

1. **New**: It activates the creation of a new program, by recalling the Window that sets the general parameters of the panel.
2. **Open**: It enters the Directory of the work's programs, permitting to recall a program that already exists in the archive: the selection procedure then follows the normal route indicated by the "My Computer" tree.
3. **Save**: It saves the current program in the default directory: the command "Save with name", which enables duplicates or savings in particular directories defined by the User, is also available from the written Menu.
4. **Print**: It prints the current program

1.3.2 General Characteristics Setting Buttons



1 2

They let you recall, at any moment, the Dialogue Box for the setting of the general characteristics of the piece, for visualization or data modify:

1. **General:** General piece data recall (Dimension, measuring units, description,...)
2. **Variable:** Recall of the **r00-r299** variables, in order to visualize or modify them

1.3.3 Current Program Handling Buttons



1 2 3 4 5

They let you operate the usual working functions on the current program:

1. **Cut:** It removes the selected parts, saving them in the note pad
2. **Copy:** It copies the selected parts, saving them in the note pad
3. **Paste:** It adds the selected parts that are saved in the note pad
4. **Erase:** It removes the selected parts, without saving them
5. **UNDO:** It undoes the last operation done



1 2 3 4

1. **Select:** It selects all the parts that are present on the current paper
2. **Find:** It searches, in the complete ISO text, for the wanted string of characters
3. **Find next:** It finds the next pattern of characters
4. **Change :** It changes the found string with the new one wanted

1.3.4 Zoom Buttons



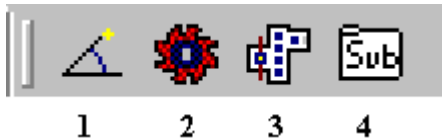
1 2 3 4 5 6 7

They let you choose and activate different kinds of Zoom:

1. **Zoom area:** It zooms on the selected area
2. **Extended Zoom:** It brings the image of the complete panel to a full screen dimension

3. **Previous Zoom:** It goes back to the previous zoom level
4. **Zoom (+) :** Enlargement following a fixed step
5. **Zoom (-) :** Reduction following a fixed step
6. **Zoom Off :** It brings the panel's image back to normal scale
7. **Pan :** Image mouse movement

1.3.5 Technology Help Buttons



While programming, these buttons let you recall data relative to the Technological Parameters of the Machine, in order to allow a check, on the congruity, for instance, of the working tools:

1. **Help Works:** It recalls the Help-on-line on the working
2. **Tool Parameters:** It recalls the Tools parameters (only for data visualization)
3. **Machine Parameters:** It recalls the Technological Parameters of the Machine
4. **Help Subroutine :** Help-on-line regarding the Subroutines

"Read Only" facilities are enabled through the access to parametric data: you are not allowed to modify them. In order to do so you have to enter, giving a password, the specific environment.

1.3.6 Graphic Simulation Buttons



It lets you recall certain visualizations, useful while programming, that show particular technological aspects of the chosen settings:

1. **Rotary Cutter Center:** It shows the real path followed by the center of rotary cutter, applying the radius correction for the selected tool
2. **Conditions:** It applies and resolves the programmed Logical Conditions (IF Blocks), hiding the parts that turn out to be not functioning.
3. **Redraw:** It redesigns the complete panel.

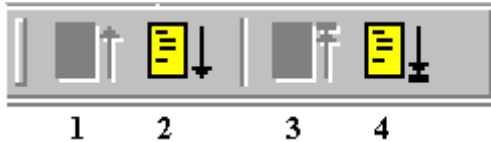
1.3.7 Block / Faces Selection Buttons



They allow you to select both the insertion mode and the current face:

1. **Upstream:** Insertion upstream of the current work
2. **Downstream:** Insertion downstream of the current work
3. **Face 5:** Face 5 selection
4. **Face 3:** Face 3 selection
5. **Face 6:** Face 6 selection
6. **Face 4:** Face 4 selection
7. **Face 1:** Face 1 selection

1.3.8 Sliding Text Buttons



They enable the sliding of the text or the direct selection of the current line:

1. **Previous Line:** It selects the line of text that comes right before the current one
2. **Next Line:** It selects the line of text that comes right after the current one
3. **First Line:** It selects the first line of text
4. **Last Line:** It selects the last line of text

1.3.9 CAD Tools Predisposition Buttons



They allow you to arrange the modalities of definition of quotes for the displacements shown in the Palette in figure:

1. **Cartesian:** by giving quotes on the Cartesian plane (X,Y e Z)
2. **Mouse:** roughly, by using the mouse
3. **Polar:** by giving the polar and angle

For what concerns the Palettes for the selection of the workings or of the design tools, please go to [Section 1.4](#) and [Section 1.6](#) respectively.

1.4 Macros-Workings

Edicad32 gives the User a set of Workings (basic level) and of defined Macros, which allow a quick definition of the working program through a simple selection, recall and application procedure of the Macros and Workings on the different faces of the panel.

Workings are archived in a particular Library, which may be configured by the Customer, by activating or not the different modalities, depending on the technological characteristics of the Machine.

Fig. 4.1 shows the possibility, among the different Configuration procedures of Edicad32, to recall

the list of Workings, either activating them or hiding them, depending on needs.

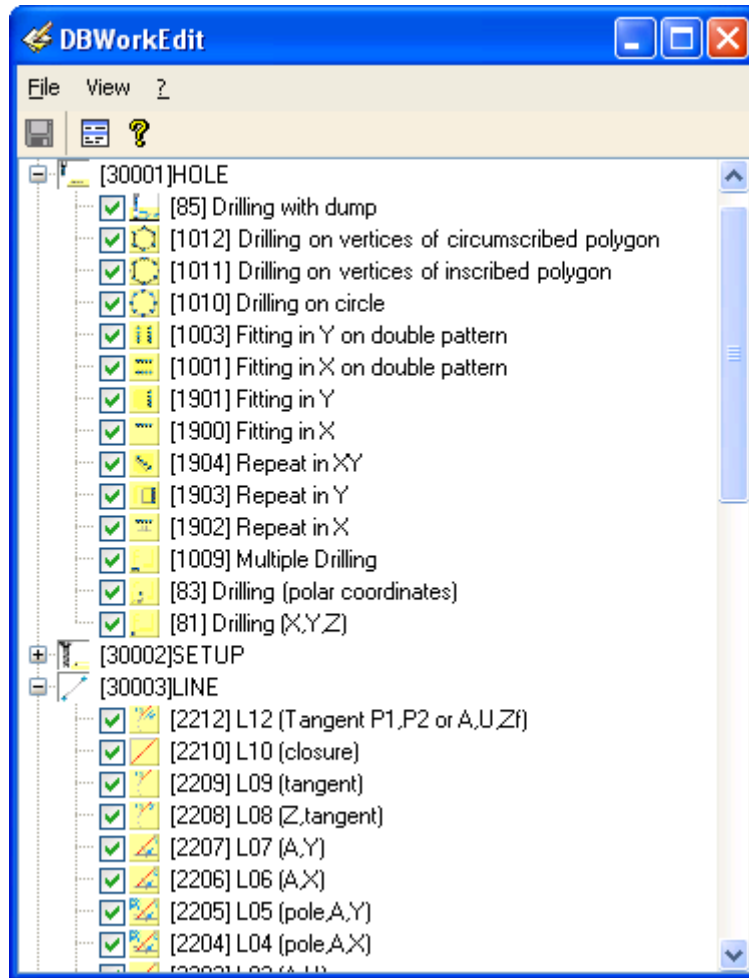


Fig. 4.1 - List of the Workings Library

1.4.1 Working Classification

What follows is a short list of the Workings and Macros usually available on the system. The detailed description of the Workings and Macros, along with some examples of how to use them, is given in [Part 2 – Workings](#) of this Handbook.

1.4.1.1 Single and Multiple Holes

Drills (X,Y,Z)	Hole (by Tool or Diameter), with application on X,Y (Cartesian) and depth Z
Drills (U,A)	Hole with Polar coordinates (module U with angle A)
Multi-tool Drills	Special Hole for multiple tool
Repeat X	Multiple Holes, in direction X, with Step and number of holes definition
Repeat Y	Multiple Holes, in direction Y, with Step and number of holes definition
Repeat XY	Multiple Holes, in diagonal direction, defined in Cartesian (Xf, Yf) or Polar (U,A)
Fitting X	Fitting on X, with step and X of Final point definition
Fitting Y	Idem, in Y direction

Fitting XY	Idem, in diagonal direction
Drills on circle	Multiple Holes on circle, defined by Center, Angle, Radius first hole
Drills on Inscribed Polygon	Multiple Holes on the vertices of a polygon inscribed in a circle, defined as the previous.
Drills on Circum-scribed Polygon	Idem, on the vertices of a polygon circumscribed to a circle, defined as the previous

1.4.1.2 Simple Rotary Cuts

Setup (X,Y,Z)	Setup rotary cutter (by T or D), with application on X,Y (Cartesian) and depth Z
Setup (U,A)	Setup rotary cutter, in Polar coordinates (module U with angle A)
Oriented Setup	Setup rotary cutter on inclined face, with angle of Pivoting
L01 (X,Y,Z)	Linear, with final Point in Cartesian coordinates (relative or absolute)
L02 (pole, U, A)	Linear, with final Point in Polar coordinates (programmed Pole)
L03 (U,A)	Linear, as the precedent, but with Pole coinciding with initial point
L04 (pole, A, X)	As L02, but with height X of the final point (instead of module U)
L05 (pole, A,Y)	Idem, but with height Y of the final point
L06 (A,X)	As L04, but with Pole coinciding with initial point
L08 (Z, tang)	Segment having a direction dictated by the initial tangent
L10	Segment of closing profile (Final point same as Setup)
A01 (Center X,Y,Z)	Arc in XY plane, having Center (I,J) and final point (X,Y,Z)
A04 (3 points)	Arc in 3 points (initial, final and intermediate) in XY plane
A05 (Center X,Z)	Arc in XZ plane, through Center and final point
A06 (Center Y,Z)	Arc in YZ plane, having Center (I,J) and final point (X,Y,Z)
A07 (3 points) XZ	Arc in 3 points, in XZ plane
A08 (3 points) YZ	Arc in 3 points, in YZ plane
A11 (R, arc min)	Arc by Radius, minor Arc, in XY plane
A12 (R, arc max)	Arc by Radius, major Arc, in XY plane
A13 (Center, A_{fin})	Arc by Center and final Angle, in XY plane
A15 (T_{gin}, Pf)	Arc having initial Tangent in continuity and final point, in XY plane
A16 (T_{gin}, Pf)	Idem, but with programmed initial tangent
A17 (T_{gout}, Pf)	Arc with programmed final Tangent and final point, in XY plane

1.4.1.3 Complex Rotary Cuts

Joint	Circular Joint on vertex, having radius R
Bevel	Straight bevel on vertex, having length U
Arc1-Arc2(C1,C2)	Double arc, with programmed centers C1 e C2
Arc1-Arc2(C1,R2)	Double arc, with center C1 and radius R2, both

Arc1-Arc2(R1,C2)	programmed Double arc, with Radius R1 and Center C2, both programmed
Arc1-Arc2(C1,Tgout)	Double arc, with Center C1 and exit Tangent both programmed
Arc1-Arc2(Tgin,C2)	Double arc, with entering Tangent and Center C2 both programmed
Arc1-Arc2(Tgin,Tgout2)	Double arc, with programmed in/out Tangents
Circle(C)	Complete circle with Center C, radius R and initial angle
Oval	Oval, given by Center, major axis, minor axis and minimum radius
Ellipse	Ellipse, given by Center, major axis and minor axis
Inscribed polygon	Polygon having N sides, inscribed in a given circle
Circumscribed polygon	Polygon having N sides, circumscribed to a given circle

1.4.1.4 Saw

Saw X	Saw in X direction
Saw Y	Saw in Y direction
Saw A	Saw in a direction given by angle A

1.4.1.5 Measures and Miscellaneous

Length Measure	Measurement cycle in X direction
Width Measure	Measurement cycle in Y direction
Thickness Measure	Measurement cycle in Z direction
Application Point	Geometric assignment of an application point for Subroutine
Offset	Offset introduction
Wait	Waiting time
Rapid (X,Y,Z)	Rapid movement with a programmed height
Message	Message transmission
IF	IF branch opening
Else	ELSE branch of an open IF
Endif	Closure of an IF

1.4.1.6 Subroutines Calls

SUB 0	Subroutine call, conditioned, single
SUB 2	Subroutine call, conditioned, repeated in X and Y

1.4.1.7 Special Functions

FOR .. Endfor	Logical block of FOR (programmed number of repetitions)
Forever	Indefinite repetition
Break	Stop
Continue	Continue
Error	Error signal
Assign. Variable\$	Value assignment to variable \$nn
Ass. var\$ oper. Ter.	Conditioned assignment with a ternary operator

1.4.2 Palettes for Working Selection

The selection and recall of a Working, to be applied step by step to the program, may be quickly done just by pointing the mouse on the corresponding icon on the Workings Palette.

The structure of the Palettes is dynamic (see Fig. 4.2): they include the Types of the main workings ("Holes", "Millings" and "Macro"); for each type a second level of Palette opens, showing the Workings available for that type.

More on the right, in the example, the specific Palette for the "Holes" type is shown.

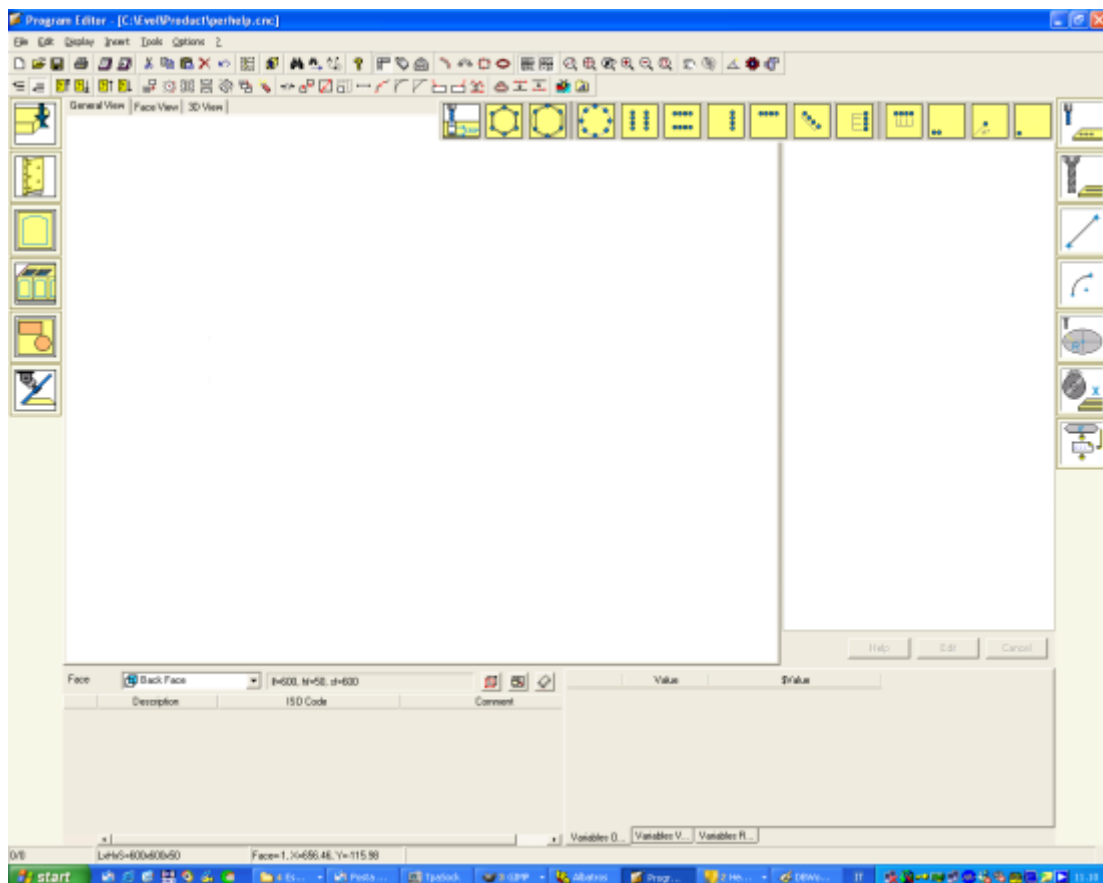


Fig. 4.2 – Working palette

1.4.2.1 Level-type Structure of the Working Palettes

Let's now give, shortly, the list of sub-levels of the Palettes relative to some of the main Workings, which are classified in the following three types:

1. Milling Workings
2. Drilling Workings
3. Saw Workings

A first level of the Toolbar Menu corresponds to this general classification, as shown in Fig, 4.3 a-b-c.

Different Toolbar sub-levels, which allow to select the following various kinds of profiles, correspond to the first group (Fig. 4.3a), characterized by the three Setup modalities (Rotary Cutter Setup, polar and inclined):

- Linear Interpolation (segments)
- Circular/Helical Interpolations (Arcs, Double Arcs and Circumferences)
- Macros for complex figures (Ovals, Ellipses and Polygons)
- Special geometric facilities (Joints and bevels)

Here below on the left we show the Workings with beside the available functions:



Fig. 4.3a
Menu regarding the workings of
rotary cutting:

**Linear,
Arcs
Double Arcs,
Ellipses & Polygons,
Joints & Bevels**

We show drilling (Fig. 4.3b) and saw (Fig. 4.3c) workings



Fig. 4.3b
Menu for the operations of
**Drilling:
Repeat Holes
Fitting
Holes on Circle and on Polygons**

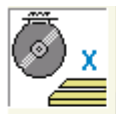


Fig. 4.3c
Saw X, Saw Y and Saw degree

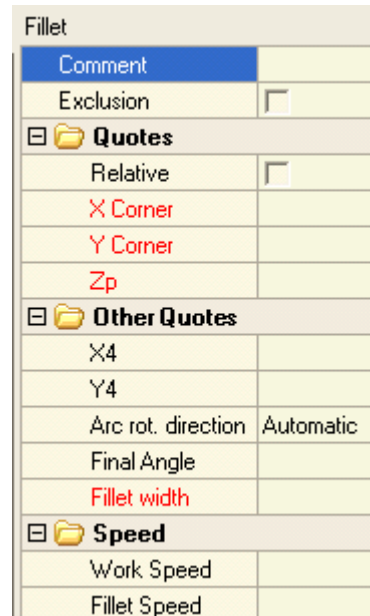


1.4.3 Data Entry Dialogue Box of Working Parameters

A Data Entry Table is dynamically associated to each recalled Working. This table helps the User during the setting of both the fundamental parameters and the optional ones, necessary to characterize the Working itself.

The table has a tree-like structure, according to the different kinds of data required, with numerical fields (they can be compulsory or optional) or check boxes inside, allowing insertion of rules for parameter interpretation.

Fig. 4.4 shows an example of such a table: the different kinds of tables will be described in detail in [Part 2 of the Handbook](#).



In the tables for the data setting, generally the fields "**Quotes**" refer to the application point of the single Workings or to the final points of each geometric entity of profile

These Quotes are usually chosen as "Absolute", that is referred to the reference system of the face, or as "Relative" (where required), that is, in this case, in an incremental manner compared to the current position, considered as the significant quote of the immediately preceding Working.

All those geometric parameters which, because of their property, have a meaning only in an absolute way (such as a radius) or other quotes that can always be defined as incremental compared to the initial point (as the Arcs' centers), make an exception.

Note that the quotes related to depths (typically Zp) have to be expressed using negative numbers, in order to follow the convention that Z+ is coming out of the panel.

Let's see in detail the general meaning of the parameters, according to the different types:

1. Parameters referred to the "Quotes" area
2. Parameters referred to the "Tools" area
3. Parameters referred to the "Speed" area

1.4.3.1 Position Area

The "Quote" parameters may obviously have very different geometrical meanings, depending not only on the kind of Working, but mostly whether it is a "Punctual" type (that is it may implicitly be represented by a point) or a more complex geometrical Entity, such as a Segment or an Arc..

The "Quote" parameters may obviously have very different geometrical meanings, depending not only on the kind of Working, but mostly whether it is a "Punctual" type (that is it may implicitly be represented by a point) or a more complex geometrical Entity, such as a Segment or an Arc..

-Drilling Position Area:

Since these are punctual Workings, the Position is referred to the application point.

Some Positions may take on particular meanings, in the case of multiple Holes or Fitting, or in the

case of having to deal with some particular parameters, such as the "deceleration positions", both in entry and exit, which will later be more precisely described along with the respective Workings.

-Milling Position Area:

Apart from the initial Setup of the Rotary Cutter (punctual too), usually the corresponding functions describe a more or less complex geometrical entity (Segment, Arc, etc.), in which the Final point is defined, implicitly assuming (for continuity) that the initial point, which isn't expressly programmed, coincides with the final point of the precedent Working (called "current Point").

Therefore the "Relative" Option may act on the Final Point, while for all the other possible parameters an unequivocal interpretation is usually applied.

For what concerns the "Angle" notation, consider that it must always be assumed that Positive = Counter- Clockwise, according to the regular trigonometric rule.

The "point hooking" (for milling profiles)

This technique (the Reader should go to the specific examples for more details on it) allows to recall and hook, at the current point, whatever programmed and previously saved profile, canceling the original Setup.

The "Multiple Setup" (for milling profiles)

This technique simplifies programming in the case of milling profiles which have to be operated, successively, using different tools, eliminating the inconvenience of having to repeat the route's definition.

-Saw Position Area

The Saw is a Working that may be executed only on face 1: the initial Setup and the final point are implicitly programmed in the instruction itself. The Reader should go to the detailed description, giving particular consideration to the aspects of the "Double Run" and of the way used to recall the "Cord Calculation".

1.4.3.2 Tool Area

This is the area used to define some of the Working's main technological characteristics, which actually consist in selecting the most appropriate Tool (or Tools) for the correct execution of the Working.

Diameter and Typology Programming

First of all let's point out that, in order to guarantee an easier and more flexible programming, most of the times the direct selection of the Tool is not a User's task, especially whenever it may be automatically chosen by the system on the basis of its pure characterization:

- As Diameter and, eventually, in addition
- As Typology, if express.

In this case the system itself will, using the tools' head configuration, automatically search for and select the Tool (or Tools) that satisfy the required specifications. In this manner the panel's programming is invariable to the specific machine's equipping. The system will obviously signal error

cases any time it is impossible to find tools fit for the required Working.

The Reader should refer to [Section 5](#) – "Equipping and Working Tools" – for the description of the various kinds of Tools that can be used for the different Workings.

1.4.3.3 Feed Area

The various tool working speeds may be set in this section of the Parameter Table, in case user would like to locally modify the default speed values, defined in the Tool Database ([→Tool Parameters](#)).

Depending on the different functions, these speeds (programmed in [mt/min]) can refer to:

1. Tool's penetration speed into the material (into the Holes)
2. Work speed, during the milling or the saw
3. Possible speed reductions in particular points (Bevels or joints), etc.

The Tools' Rotary speeds are kept separate from these; they are also locally programmable and are expressed in [rpm].

→ Note: Where not programmed, the system automatically assumes the default speeds, which are expressed in Parametric form, associated with the selected tool.

Drilling Speed Area

The Drilling Speed Area consists only of two fields to fill in:

- **Entry Speed:** referred to the tool's penetration movement (the exit is executed in rapid)
- **Spindle Rate Number:** referred to the Spindle rotary speed.

As already stated, it is possible to program some pieces, along the Hole, where tool's entry speed reductions previously defined may be applied.

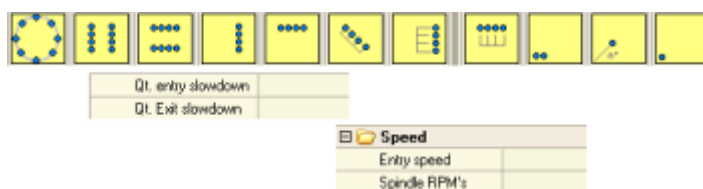


Fig. 4.5

In the drilling speed area, you must keep in mind the possibility to require specific decelerations, both in entry and exit of the passing hole.

Milling Speed Area

During the milling Working, it is possible to locally program the speeds (tangent) of the single lengths: the value is automatically spread to the lengths that immediately follow, up to the programming of a new value. In the complex Macro-instructions (as Ellipses) both the entry and the work speeds are required.

For the particularly complex points, as the joints, a speed reduction is explicitly suggested, because of possible quick changes of the axis direction:

☰ Speed	
Entry speed	
☰ Speed	
Work Speed	
Fillet Speed	

- **Working Feed:** tool's usual work speed

- **Bevel/Round Feed:** possible reduced value of the work speed.

Saw Speed Area

In this specific case the Entry, First Pass and Second Pass Work speeds are pointed out and may be programmed one by one.

☰ Speed	
Entry speed	
Work Speed	
Speed of second pass	
Double pass	<input checked="" type="checkbox"/>

Fig. 4.6

In the saw working it is possible to program the second pass speed separately from the speed in the first

1.5 Outfits and working Tools

1.5.1 Working Tools programming

1.5.1.1 Programming by Diameters and typology

In the Workings programming, Tool may be selected or directly - by the corresponding Spindle number entering - or implicitly (in the "Drillings" case), by the "Diameter" and, optionally, the Typology requirements.

In the case of direct selection, we can see the programming rules and the available Help for easiest Tool identifying.

1.5.1.2 Drilling Tools

Concerning Drilling operations, different Tool selection criteria may be assumed: for every drillings typologies the Group and Tool number are required:

Group: some Tool-machines may be equipped with two different Tool-Heads (eventually with the same tools nomenclature): in this case a "Group" number is required for the Head selection (otherwise, a default number Group = 1 is assumed).

Tool: program attributes a sequence progressive number to every tool-spindle, in an univocal mode. In effect, the "tool" field value identifies the corresponding spindle position.

To verify the available current tools configuration, Programmer may recall the "Help on line" command:

- **Outfits**

that allows to visualize the current Tools disposition on the multi-spindles Head.

The following Table summarizes the selection rules of the tools, for the most important drilling typologies:

Automatic	Yes	-	-	-	-	-	-	-	-	-
Diameter	Yes	-	Yes	Yes	Yes	-	-	Yes	Yes	Yes
Group	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Tools	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Typology	Yes	-	-	-	-	-	-	Yes	Yes	Yes

Fig. 5.1

In this Table are listed the required information for direct Tool selection.

Obviously, in the case of "Programming by Diameter", only this last, and, eventually, the Typology, are required.

More precisely:

Automatic: if this option is selected, no particular values are required in the Tools fields, since program will provides automatically to select the most convenient Group and Tool.

Diameter: this parameter will be used as the most important "searching key" for the tool-hole matching.

Typology: This is an optional parameter, may be used added to diameter, as auxiliary attribute for matching. Normally typologies include: *Blind Drilling tool*, *Crossing Drilling tool*, *Drilling with Saw tool*, *Drilling with Flare*, and so on.

Some Drilling typologies cannot be recalled in all the working faces; in the following Table we can see the compatibility conditions:

		Blind Drill 	Crossing Drill 	Drill & Saw 	Drill & Flare
Face 1		Yes	Yes	Yes	Yes
Face 4 and 6		Yes	No	No	No
Face 3 and 5		Yes	No	No	No

Some Data Entry tables, for tools selection, are showed in the following examples (Fig. 5.2 a-b-c-d).

Tools	
<input checked="" type="checkbox"/> Automatic	
Diameter	
Group	1
Tool	1
Typology	

Fig. 5.2a

Automatic Option selected.

By default program select the Group = 1 and the Tool = 1 .

Tools	
<input type="checkbox"/> Automatic	
Diameter	
Group	1
Tool	8
Typology	

Fig. 5.2b

In this case, **Tool** has been directly selected (by the Group and Tool numbers).

Tools	
<input type="checkbox"/> Automatic	
Diameter	8
Group	1
Tool	
Typology	1

Fig. 5.2c

In this case, Tool is required with the parameters: **Diameter = 8** mm and **Typology = 1** (Blind drilling tool).

Tools	
<input type="checkbox"/> Automatic	
Diameter	8
Group	1
Tool	
Typology	

Fig. 5.2d

In this case only the **Diameter** value has been programmed: program will select the first tool detected with Diam=8, without considering typology.

1.5.1.3 Milling tools

The following Table summarizes the milling Tools characteristics. A special "**Tandem**" execution provides two contemporaneous milling cycles, then with the selection of a milling tools pair.





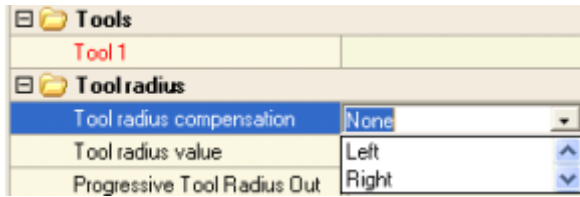
	Mill Setup 	Mill Setup Polar coordinates 	Oriented Mill Setup 	Oval, Ellipsis, Polygons 
Automatic	Yes	Yes	No	No
Group	Yes	Yes	Yes	Yes
Tool 1	Yes	Yes	Yes	Yes
Tool 2	Yes	Yes	Yes	No
Mill Radius value	No	No	No	Yes
Mill radius Correction	No	No	No	Yes

Fig. 5.3

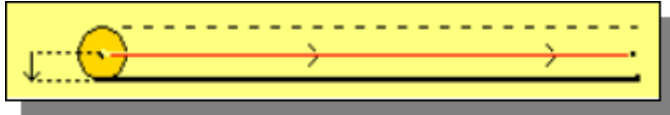
For the different executive typologies (in the columns) the different using levels (for options and parameters) are showed.

- **Automatic:** the Automatic option, if enabled, forces the program to select the Group and Tool defined by default. Normally this option is used in the case of DXF files importing. In this case the tool fields are not accessible!
- **Group:** When not in "Automatic" mode, the Group code is compulsory (if machine is configured with only one Tool-head, the Group=1 is assumed by default).
- **Tool 1:** The Tool number programming is compulsory, **except** in "Automatic" mode. "By Diameter" programming is not allowed in the Milling works.
- **Tool 2:** Must be programmed only in the case of Tandem execution.
- **Mill Radius value:** This parameter is not used for an automatic tool selection, but only for the Tool Radius Correction procedure (see next parameter). Programmer must consider that this value is used for the Tool Center path computing: consequently, if programmed with a different value (respect the effective Tool Diameter, defined in the "Tools Parameter") may produces a special milling execution as, if entered with a greater value, leaving in the piece an extra-thickness (convenient in the rough steps).
- **Mill Radius Correction:** If enabled, requires also the Correction side (left or right), relative to the moving sense. Mill radius correction assumes, as parameter, the previously entered value and computes the effective toll center trajectory.
- **Option: Progressive Radius correction** - If this option is enabled in the Setup, this initial position will not be corrected while correction will be computed on the final point of the first profile bloc. The system will generate a special moving instruction, from the Setup point (not corrected) to the end point of the first bloc (corrected).

In the following, the different Mill Radius Correction procedures are showed, in terms of programming modes and executive result (see Fig. 5.4a-b).

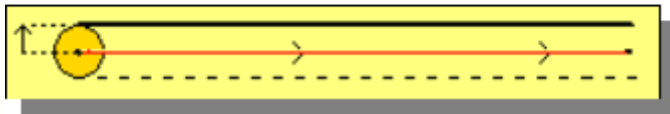
**Fig. 5.4a**

The Mill Radius Correction is required on "Left" side, with Radius = 5 mm. If the Radius value is missing, program will assume the "Tool Parameter" assigning.

**Fig. 5.4b**

In the upper figure, the MRC procedure is selected in "Left" side, in the lower in the "Right" side.

In Edicad32, the special command button :

**Direction visualizing**

allows to enhance, in the drawing, the moving sense

In the following Table, the different Milling Tools typologies and the Working Faces compatibility are listed:







	Shaped Mill 	Rotating Head 	Smoothing Mill tool 	Vertical Mill 	Rotating Head with Offset 	Toroidal Mill 
Face 1	Yes	Yes	Yes	Yes	Yes	Yes
Face 4 e 6	No	Yes	No	Yes	Yes	No
Face 3 and 5	No	Yes	No	Yes	Yes	No

Fig. 5.5

As showed in the Table, some particular milling Heads cannot be used on every faces: only in the face #1 all typologies are allowed.

In programming, the different Mill typologies, supplied in the Technological Data Base may be recalled with the Help command:

Tool Help-on-line:

**1.5.1.4 Sawing Tools**

The Sawing working do not requires particular building functions, since the path may be only based on a single segment: horizontal (Saw X), vertical (Saw Y) or along a sloped straight line (Saw degrees).

Then a single instruction may code as its Setup (Entry and initial point) as the Final point.

Nevertheless, the special working and tool characteristic cause a particular management for the "Mill Radius Correction" mechanism and for the "Chord correction".

In particular:

Mill Radius Correction: in this case, this procedure assumes the special meaning since correction is related to the **Saw Thickness**, that is considered (as the diameter for the Milling case) to compute the effective tool Center path.

Saw Diameter: The Saw diameter is also considered, but for another correction procedure (the "**Chord correction**" , described in the [Part 2- Edicad32 Workings](#)) may be recalled to compensate the groove enlarging effect, caused by the tool diameter, then reducing the effective tool center movement length.

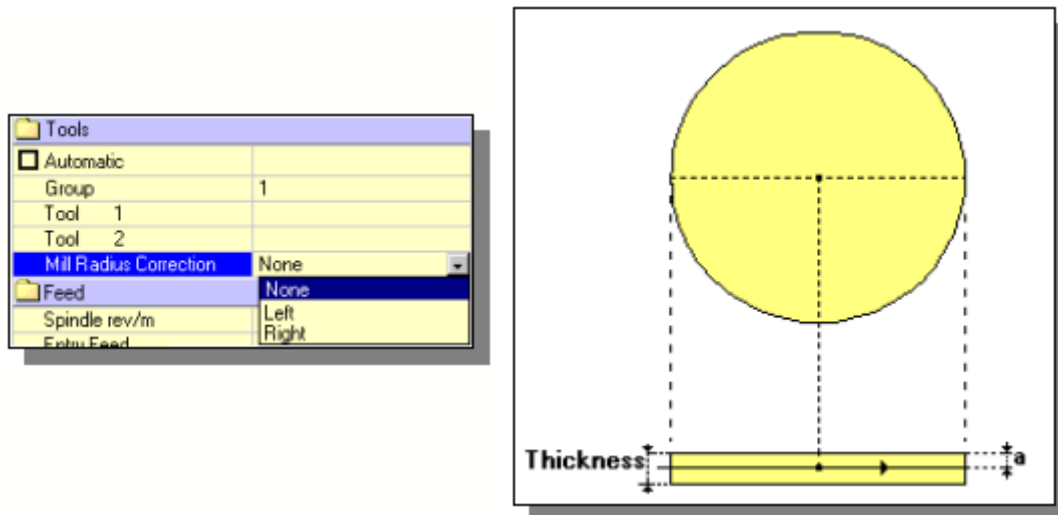


Fig. 5.6

In the figure is enhanced the effective a parameter (half-thickness), considered for the Tool Radius Correction procedure.

1.5.2 Help on Line for Tools


The "Help on line facility allows to recall, in the Edicad32 environment, all the information included in the Technological Data Base as, for instances, the current machine Outfits, the available Tools characteristic, in order to verify, in programming, the technological compatibility between workings, faces and selected tools.

The Technological Database may be accessed, from Edicad32, in "Read only" mode: all data may be recalled for displaying but cannot be modified.

To recall the "Help for Tools" facility, the following command-button must be selected:

- Help for Tools



or it is possible to click on the **Tool** field button  on the working setting dialog box

In the corresponding graphic windows, Programmer may verify if the selected tool is "compatible or not" with the current working: the following icon are displayed on the side of the tool identifier:

An important aid for tool selection is provided by the information included in the **ID** field, which shows the tools which can be selected for the current working

The most common situations of incompatibility occur:

- when the selected tool is not enabled for the required face, as, for instance, when "Smoothing Mill" are attempted on the faces 3,4,5 and 6
- when the selected tool is not compatible with the required working typology.

Fig. 5.7 - Help on Tool

In the Fig. 5.7 and 5.8a-b some examples of "Help on Tools" are showed, with the graphic images may be recalled from the Technological Database.

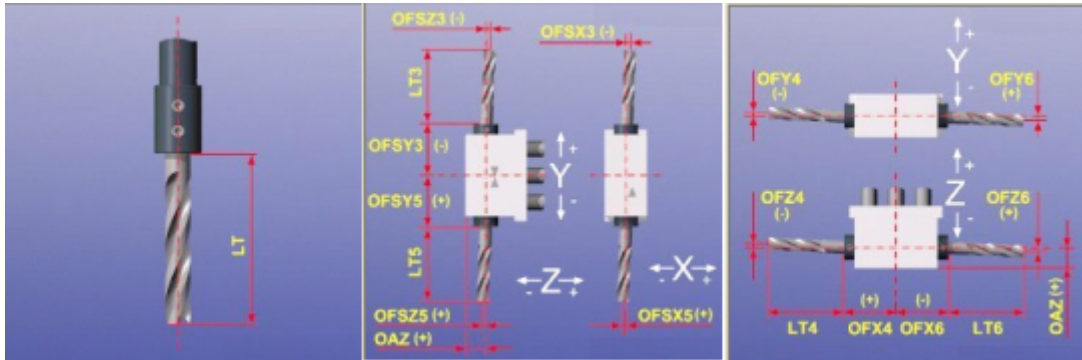


Fig. 5.8a

Examples of "Help on Tools" images, recalled from Data Base, and related to the Drilling tools working on faces 1, 4 / 6 and 3 / 5.

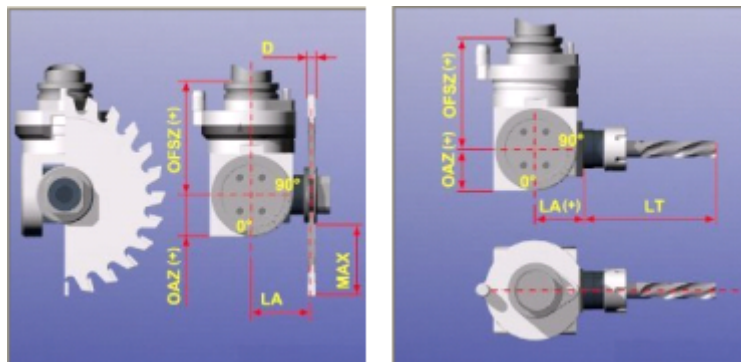


Fig. 5.8b

Rotating Heads, with Sawing and Milling tools.

1.6 Drawing Manipulation

Edicad32 provides a very powerful Drawing environment, based on Graphic Manipulation commands set, designed to allow rapid workings modifying or multiplying. These commands usually work on a single Macro-work or on a workings group or yet on entire milling profile, according to the selection mechanism previously used.

All these Drawing commands may be run from the Literal menu or, most rapidly, directly from a CAD tools Palette.

1.6.1 CAD manipulation tools Palette

On the current program different graphic tools allow to manipulate the pattern geometry or the milling profile characteristics.

Before the tools recalling, as later listed and described, Programmer must identify, selecting them, all the geometrical objects will be subjected to manipulation and more must define (using the special commands listed as 1 to 3, in the Sect. 1.3.9) the Reference mode (between Cartesian, with Mouse or Polar) for the quotes and target positions of the operation will be recalled.

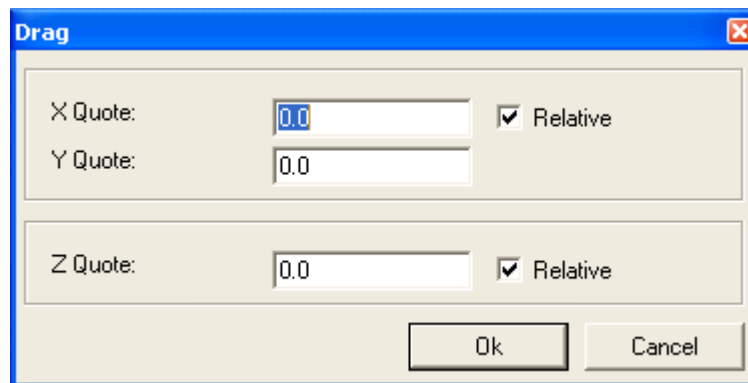
When these rules have been defined, the recalled Command will work on the selected geometry.

1.6.1.1 "Move" command

It's recalled by the button :



It works as on the "single point" Workings (as Drillings) as on the Profiles: with this command, all the selected workings are moved to the new programmed X, Y and Z quotes, defined in Absolute or Relative modes.

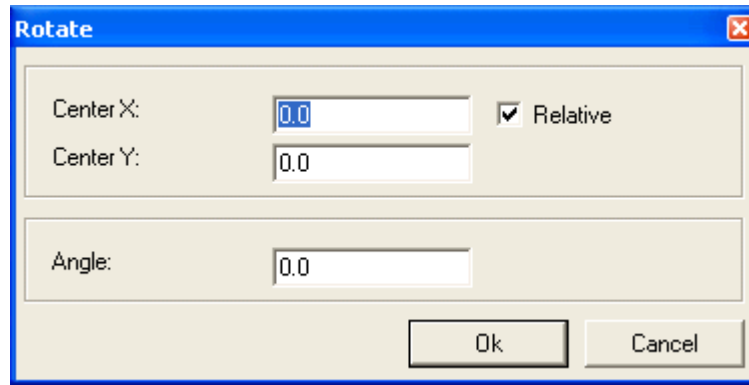


1.6.1.2 "Rotate" command

It's recalled by the button:



This operation, normally used for profiles, allows to rotate the drawing, by a programmed angle, assuming as center a defined point, programmed in absolute or relative (to final point) mode.

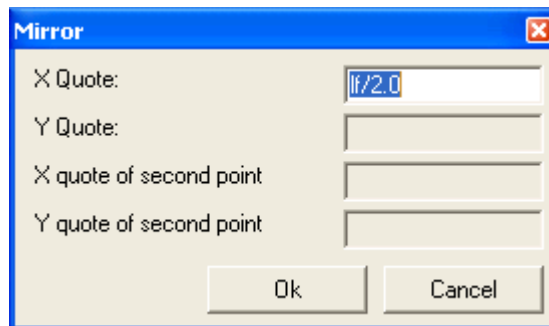


1.6.1.3 "X Symmetry" command

It's recalled by the button:



This operation allows to duplicate the selected workings, mirroring them respect to a vertical symmetry axis, positioned to a programmable X quote (by default proposed in the medium of the panel: $x = L/2$).

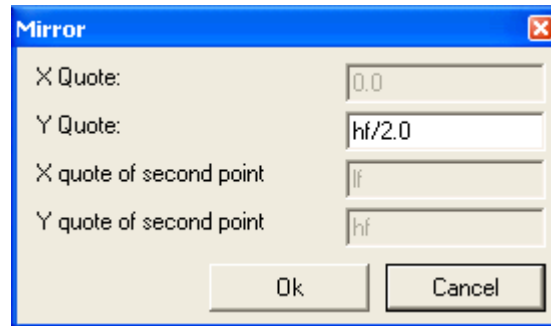


1.6.1.4 "Y Symmetry" command

It's recalled by the button:



This operation allows to duplicate the selected workings, mirroring them respect to an horizontal symmetry axis, positioned to a programmable Y quote (by default proposed in the medium of the panel: $h = H/2$).

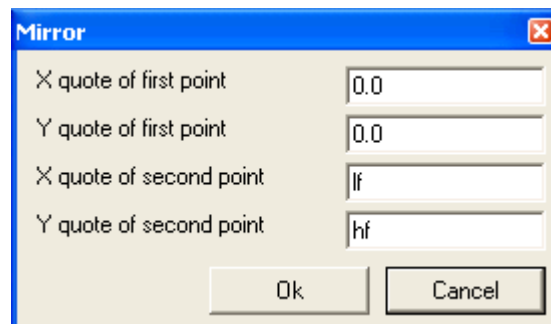


1.6.1.5 "X/Y Symmetry" command

It's recalled by the button:



This operation allows to duplicate the selected workings, mirroring them respect to a sloped symmetry axis, defined by a Points pair.

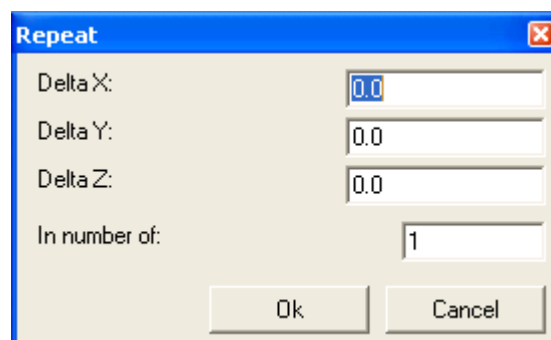


1.6.1.6 "Repeat" Command

It's recalled by the button:



This operation allows to repeat more times the selected workings, moving them along a sloped direction, defined by the three components : Delta X, Y and Z.



1.6.1.7 "Explode" command

It's recalled by the button:



This command allows to "divide" the selected profile or subroutine in the single geometrical elements.

1.6.1.8 "Profile cutting"

It's recalled by the button:



This command allows to "cut" the current profile in correspondence of the selected internal point: this point will assume the Setup meaning for the second part of profile.

1.6.1.9 "Profile reversing"

It's recalled by the button:



This command allows to reverse the moving sense along the selected profile, then causing exchange between the original Setup and Final points.

1.6.1.10 "Move profile Setup" command

It's recalled by the button:



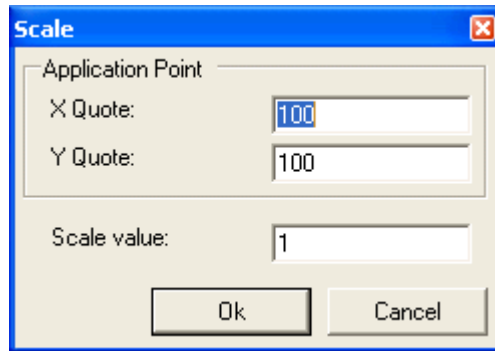
This command may be referred only to a closed profile: it allows to move Setup from original to a new programmed internal point: the moving sense don't change.

1.6.1.11 "Profile Scaling" command

It's recalled by the button:



This command allows to increase or reduce the selected profile, applying to them a programmable Scale factor



1.6.1.12 "Profile Link" command

It's recalled by the button:



This command may be referred only to two different profiles, one of these assumed as "first", the other as "second": command allows to "move" the second profile as its original initial point (Setup) **coincides** with the final point of the first: the second Setup will be erased, causing the two paths linking.

1.6.1.13 "Profiles connection" command

It's recalled by the button:



This operation is similar to the "[Profile Link](#)" command but don't require the second profile moving: the original paths linking is performed by a segment traced from the final point (of the first) to the Setup point of the second.

The second Setup operation is erased.

1.6.1.14 "Round" command

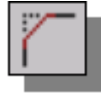
It's recalled by the button:



If one or more polygon Vertices have been selected, this command allows to round off them with a **programmable Radius**.

1.6.1.15 "Beveling" command

It's recalled by the button:



If one or more polygon Vertices have been selected, this command allows to insert a linear "beveling", with a **programmable Distance** from the corner.

1.6.1.16 "Entry path" command

It's recalled by the button:



When a milling profile, and its Setup point are selected, a programmable "**entry path**", linear or circular with tangent continuity (relatively to the profile initial direction) may be inserted with this operation: the Length (for "linear") or Radius (for "circular") values may be programmed. The original Setup is moved in back to the initial point of the added path.

1.6.1.17 "Exit path" command

It's recalled by the button:



Similar to the previous command, allows to add (to the Final point of the selected profile) a linear or circular "exiting path", with tangent continuity with the final direction.

1.6.1.18 "Z Laying"

It's recalled by the button:



Starting from a selected profile, that presents changes in Z quote, this command allows to generate a new path, "**projected**" on the XY plane, where the Z values are annulled.

1.7 Parametric Programming

The Edicad32 system provides a very powerful "Parametric Programming" environment, expressly designed to supply to the final User logical facilities and tools to simplify the part programs development, reducing costs and time.

Parametric programming allows to answer to the flexibility and modularity requirements, typical of the Application, where the panels working layouts need to be rapidly modified and customized, following well defined geometrical rules, related to the constructive concepts of the composite furniture.

The basic concept is that is possible to identify, for every panel typology, a standardized working

layout just including, in its own description, all the mathematical and logical rules able to generate, automatically, all the possible variants may be required in all the special executions (for change of nominal dimensions or customizations) may occur in production.

In practice, this methodology allows to define the working program, for the "standard" panel, in terms of a set of basic instructions, which structure (in terms of geometrical characteristics) may change on the basis of mathematical expressions, including a set of "variables" that may code the specific execution conditions.

According to these basic concepts, all the working Macro-instructions, included in this basic program, will be characterized by a set of variable "parameters" and mathematical formulas that allow to modify the final result simply re-assigning their values with the required current data.

This mechanism may also be enhanced using a "Parametric" Subroutines Library: with an appropriate subroutines organization, every part program may be realized in terms of a set of subroutines recalls, each of one with their internal parameters re-assigning with the current values, required for every special execution. The parametric subroutines organization allows to reduce the total quantity of different executive programs must be stored in the archives, since special executions may be easily derived from the standardized types, only by a "local" parameters re-assigning.

Practically, the "Parametric Programming" concepts suggest User to follow these logical steps:

1. In the beginning, analyze, for every panels typology, the typical working patterns, in order to identify the most recurrent layouts may be saved in form of subroutines
2. Then recognize, for these "standard" recurrent layouts, the logical and geometrical rules that may change their structure according to special requirements, as, for instance, when changes of the main panel dimensions are required
3. At this point, define the corresponding mathematical expressions, including the dimensional data (or other significant Variables), may code these modifying laws, and use these expressions to represent all the significant parameters (quotes, distances, angles, and so on) that may characterize the geometrical pattern of the workings

Finally, organize the Main Programs in terms of successive subroutines recalling, passing to these the effective current parameters values, as required for every specific execution. The main program itself may be characterized in terms of a set of internal variables, may be every time re-assigned, in order to generate, when required, particular variants according to special execution requirements.

1.7.1 Variables

The Parametric Programming modality, supplied by the Genesis Evolution Numerical Control, is based on the possibility to define and recall a set of program "Variables" and mathematical expressions to code the Workings parameters.

System provides three different Variables typologies:

Offset Variables: this variables group may be used to define local translations of the Reference Origin of the Axes, then allowing more flexibility in the Machine setting and in the panels arrangement on the working plane.

	Value	\$Value
Offset X	0	
Offset Y	0	
Offset Z	0	

Fig. 7.1

The Offset values indicate the local translation may be associate to the panel, when positioned differently to the usual Field Reference pins. (Fields Origins).

"V" variables : Include 8 special program variables, not directly used in the mathematical expressions, that may be transferred to the PLC program, for the Machine Interface (Fixed Cycles) customization.

	Value	\$Value
v0	100	100.0
v1	50	50.0
v2	0	0.0
v3	0	0.0
v4	0	0.0
v5	0	0.0
v6	0	0.0

Fig. 7.2

These Variables are directly managed by the PLC program.

"R" Variables : This is the most important program Variables class (until 300 variables are available, named **r0** to **r299**) since may be locally defined in any Main program or Subroutine and used in the formulas for the Workings parameters defining.

	Value	wr	Format	\$Value
r0	100		Float	100
r1	50		Float	50
r2	-20		Float	-20
r3	trefori		String	trefori
r4	0		Float	
r5	0		Integer	
r6	0		String	

Fig. 7.3

The "R" Variables scrolling Data Table (r0 to r299), for displaying and programming.

The "R" Variables may be used for numerical values or characters string coding, since provide the following Data type format:

- Integer**: for integer numerical value
- Float**: for numerical values in Floating point
- String**: for an ASCII characters string variable.

The R-type Variables may be also used to define the "Offset" and "V" Variables.

The R-type Variables may be marked as "Re-assignable" (field w=[x]), as, typically, in the case of the Subroutines, where may be required to be modified by the recalling program.

1.7.1.1 Variables assigning

The R-type Variables must be assigned, as Value and Typology, at the Program opening, before to be recalled and used in the mathematical formulas for the workings parameters definition.

Then, any rxx variable, will be supposed for using, must be preventively initialized with a numerical value. Alternatively, any variable may be itself defined in parametric format, by a mathematical expression including other rxx variable, provided that only already defined and with lower

progressive number rxx are used.

Practically, it's required that any mathematical formula must be resolvable and computable.

In the case of Subroutine, at the creating moment, its internal Variables must be numerically defined, as to allow a local interpretation. All of these that are supposed will be modified by the recalling Main program, must be countersigned as "re-assignable", setting the field **w** =[x].

1.7.1.2 Program basic Variables

In any program the following internal "basic" Variables are implicitly defined and available for using:

-L, H, S: dimensions of the piece: these variables are automatically defined in the "Piece Definition" session (see Part [Ex. 1.0](#) of the present Manual). These variables may be used, as numerical value, in any field of the Working Parameters data tables.

-Lf, Hf, Sf: dimensions of the current working Face (logically related to the previous).

We can see, for instance, as the "medium" point of the Face #3, for a panel with dimensions [600x400x80], may be indicated :

1. By a direct numerical coordinates entering, then $X = 300, Z = 40, Y = 0$.
2. By the L, H, S variables (piece dimensions), with the expressions : $X = L/2, Z = S/2, Y = 0$.
3. By the lf, hf, sf variables (Face dimensions), with the expressions: $X = lf/2, Z = hf/2, Y = 0$.

→ Note: This is a lateral face of the panel, so the variable "hf" is logically associated to the panel thickness, therefore it refers to the Z axis direction.

1.7.2 Mathematical Expressions and Operators

In the Parametric Programming, all expressions include the Variables and a set of mathematical Operators and Functions using, with parenthesis "nesting", considered conforming the usual priority arithmetical rules.

Expressions allow to correlate the recalled Variables with a set of basic Arithmetical and Mathematical (Transcendent) Operators and Algebraic and Trigonometric Functions.

1.7.2.1 Basic Arithmetical Operators

Symbol	Operation	Syntax
+	Addition	$4=r1+r2+r3$ or $r4=r4+7$
-	Subtract	$r5=r4 - r8$ or $r5=r4 - 3$
*	Multiply	$r5=r2*r4$ or $r5=r4*3$
/	Division	$r6=r5/r2$ or $r5=r2/7.89$
#	Integer Part	$r6=r9\r7$ returns the Quotient Integer Part
%	Remainder	$r7=r5\r4$ returns the division Remainder
?	Step adjust	$r7=r6\r4$ returns Divisor modified in order to produce an Integer result of the Division
	Logical OR	Returns the logical OR of the Integer Part of the Operands
&	Logical AND	Returns the logical AND of the Integer Part of the Operands

→ Note:

1. Division may return a decimal value, according to the operands original values
2. The Integer Part operation realizes a decimals cutting off on the Quotient (result of division), returning an Integer value, eventually approximated by defect.
3. The Remainder operation provides, if needed, a preventive cutting off of the decimals of the two operands of the division (then reducing them to an integer numbers), then returns the value of the Remainder.

Examples:

Assuming $r2 = 27.15$ e $r5 = 4.73$ as current values for the Numerator and Denominator:

- a) **Division:** $r6=r2/r5$ $r6$ assumes the value of **5.7399577..**
- b) **Integer Part:** $r6=r2\r5$ $r6$ assumes the value of **5**
- c) **Remainder:** $r6=r2\r5$ preventive **Cut off** of **r2 to 27** and **r5 to 4**
r6 assumes the value of **3** ($27 / 4 = 6$ with the Remainder = 3)

1.7.2.2 Mathematical transcendent Operators

The following mathematical transcendent Operators are available:

Symbol	Operation	Syntax (Argument cannot be an expression)
abs[.]	Absolute	r2=abs[r0] Returns the Absolute Value of the Operand
sqr[.]	Square Root	r4=sqr[L1] Returns the Square Root of the Operand
pow[.]	Power	r5=pow[L4] Power of 2
int[.]	Integer	Returns the truncated Integer Part of the Operand
inv[.]	Reverse	Returns the Reverse (1/x) of the Operand
round[.]	Rounding	Returns the nearest Integer value
gr[.]	Degrees	Converts Operand from Radiant to Degrees
odd	Odd	Returns 1 if the Integer Part of the Operand is Odd (0 if Even)
strlen[\$.]	Lenght	Returns the characters String length

→ **Note:** The "Square root" argument must be a positive number

Examples:

Square Root: **r5=sqr[712.25]** **r5** assumes the value of **26.6880122..**

Power of 2: **r5=pow[r4]** supposing **r4 = 4.25**, **r5** assumes the value of **18.0625**

1.7.2.3 Trigonometric Functions

The following Trigonometric Functions are available:

Symbol	Operation	Syntax (Argument cannot be an expression)
sin[.]	Sine	r4=sin(r5) o r4=sin(30) argument in [degrees]
cos[.]	Cosine	r2=cos(r6) o r2=cos(-45) argument in [degrees]
tan[.]	Tangent	r3=tan(r7) o r3=tan(80) argument in [degrees] (*)
asin[.]	Arcsine	Reverse function of Arcsine (argument from -1 to 1)
acos[.]	Arcocosine	Reverse function of Arccosine (argument from -1 to 1)
atan[.]	Arctangent	Reverse function of Arctangent (returned from 0 to 180°)

→ **Warning:** The "Tangent" function may generate Overflow errors with argument = +/- 90°

Examples:

a) Sine: **r4=sin(30)** **r4** assumes the value of **0.5**

b) Cosine: **r2=cos(-45)** **r5** assumes the value of **0.7071067..**

c) Tangent: $r3=\tan(80)$ $r5$ assumes the value of **5.6712818..**

1.7.2.4 Special and Technological Functions

These special Functions, normally not available for the Final User, allow a special Macro realizing.

getat [nn;np]	Returns the decimal value of the character np extract from the \$nn string
pown [nb;ne]	Generic exponent Power (nb = base ; ne = exponent)
hypot [c1;c2]	Returns the Hypotenuse, with c1 and c2 as catheti
ifelse [nc;n1;n2]	Returns n1 if nc#0 or n2 if nc=0
ifcase [nc1;nsp;nc;n1;n2]	Evaluates the condition (nc1;nsp;nc2) and returns n1 if True, n2 if False
case [nc;nc1:nv1;nc2:nv2..]	Multiple Test: returns nv# corresponding to the nc# equal to nc
min	Minimum between a max of 30 operands
max	Maximum between a max of 30 operands
ave	Average of max 30 operands
sum	Sum of max 30 operands
prface	Test for compatibility between Tool and Face
prtool	Returns the required parameter, related to the selected Tool
primp	Returns the required parameter, related to the Plant
pronegr	Returns the required parameter, related to the Group of the Plant
prmac	Returns the required parameter, related to the Machine

For more details, please contact the manufacturer.

1.8 Parametric Subroutines

In the panels working, very frequently occurs that similar drilling layouts are recurrently required, with little differences, for panel with the same typology but different dimensions or customizations. Very frequently, these drills patterns change according to a repetitive geometrical rules, as defined by the furniture composition.

According to these general requirements, the Edicad32 programming environment allows to compose the working programs in terms of aggregations of basic subprograms (named "Subroutines", in the following), that may be previously created and saved.

In order to be used in the most possible generalized mode, Subroutines may be created in a "parametric" format: their internal workings structure may be based on a set of re-assignable variables, that will be defined by the recalling program. For instance, the internal drilling pattern may be designed as to adapt itself to the panel dimensions, as defined in the main program.

System is structured in very powerful mode, since provides until 5 "nesting" levels of subroutines, allowing a very complex parametrical rules.

Then the final User has the possibility to analyze the typical requirements of own panels typology, in order to identify the most recurrent drilling schematics, in the most elementary form, and to create the basic subroutine level; then, passing to the following levels, to aggregate this as to obtain the most complex layout, introducing the parametric rules for composition.

Obviously the "parametric and structured" programming mode allows to simplify the normal programming phases, when the subroutines set has been prepared, since the current programs will be reduced to a simple pre-built subprograms aggregating, with auto-adapting rules already defined and tested, then reducing the programming time and errors.

Edicad32 supposes a structured archives, where executive "main programs" and "subroutines" are saved in different dedicated subdirectories:

- The Main programs, in the "**Product**" directory ,
- The subroutines in the "**Product \ sub**" subdirectory.

No differences exist, between programs or subroutines, concerning the usual programming rules and modalities: only, a program save in the main directory cannot be recalled as subroutine.

1.8.1 Subroutines recalling functions

Edicad32 supplies two different recalling modes, the SUB0 and the SUB2 functions, which Data Entry table is showed in the Fig. 8.1.

These two function differs only for the "multiplying" mechanism. This last procedure allows to recall more times, in contemporary, the same subroutine, changing, with a programmable rule, the application point:

- by **SUB0** : multiplying is realized along a programmed **sloped line**, where the different application points are defined by the components (the "Offset X" and "Offset Y" parameters) of the distance: more a progressive change of depth may be programmed with the "Offset Z" parameter
- by **SUB2** : multiplying is realized on a **rectangular pattern**, characterized by a programmable number of "rows" and "columns", and their inter-axes, also indicated as "Offset X" and "Offset Y".

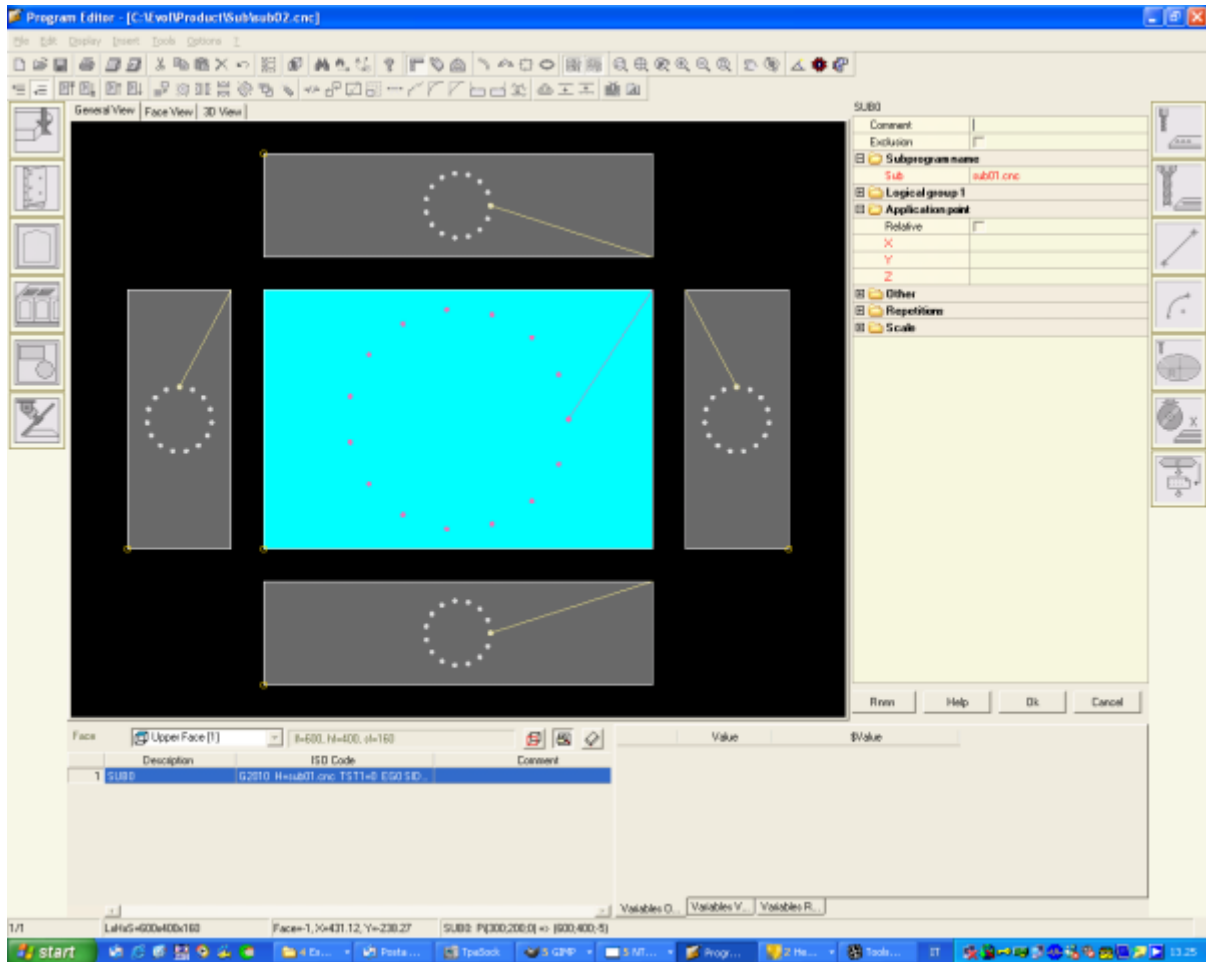


Figura 8.1 - Example of a Subroutine recalling on all the 5 faces.

1.8.1.1 Recalling Options: Translating, Rotating and Mirroring

In the subroutines recalling, many manipulation optional commands (will be described in the Part 2 - Sect. 5 of the present Manual) allow to modify the application logic or the internal pattern as, synthetically:

- import, in the main program, the complete subroutine or only a "single face" part: then, in the internal subroutine body provides working on different faces, only the selected face will be imported and inserted on the current main program face (that may equal or different to the recalled
- then, according to this last possibility, insert in the main program, working originally programmed (at level of subroutine) on another face: then, for instance, a particular drilling pattern may be programmed, in the subroutine, only one time and on the basic face, but recalled more times and inserted in more different faces of the main panel.
- With a single recalling function, to "repeat" application along a programmable pattern of "knots".
- Translate and/or rotate the subroutine, when inserted
- Hook them to the "current" point (very convenient for milling profiles)
- Mirror the subroutine itself, with an X or Y symmetry operation
- Increase or reduce its internal logical dimension, according to a programmable "Scale factor"
- Recall them, with its internal Variables re-assigning, in order to modify layout and structure
- Force a Logical Condition for recalling, defining a "Comparing relation" based on mathematical expressions including the main program Variables.

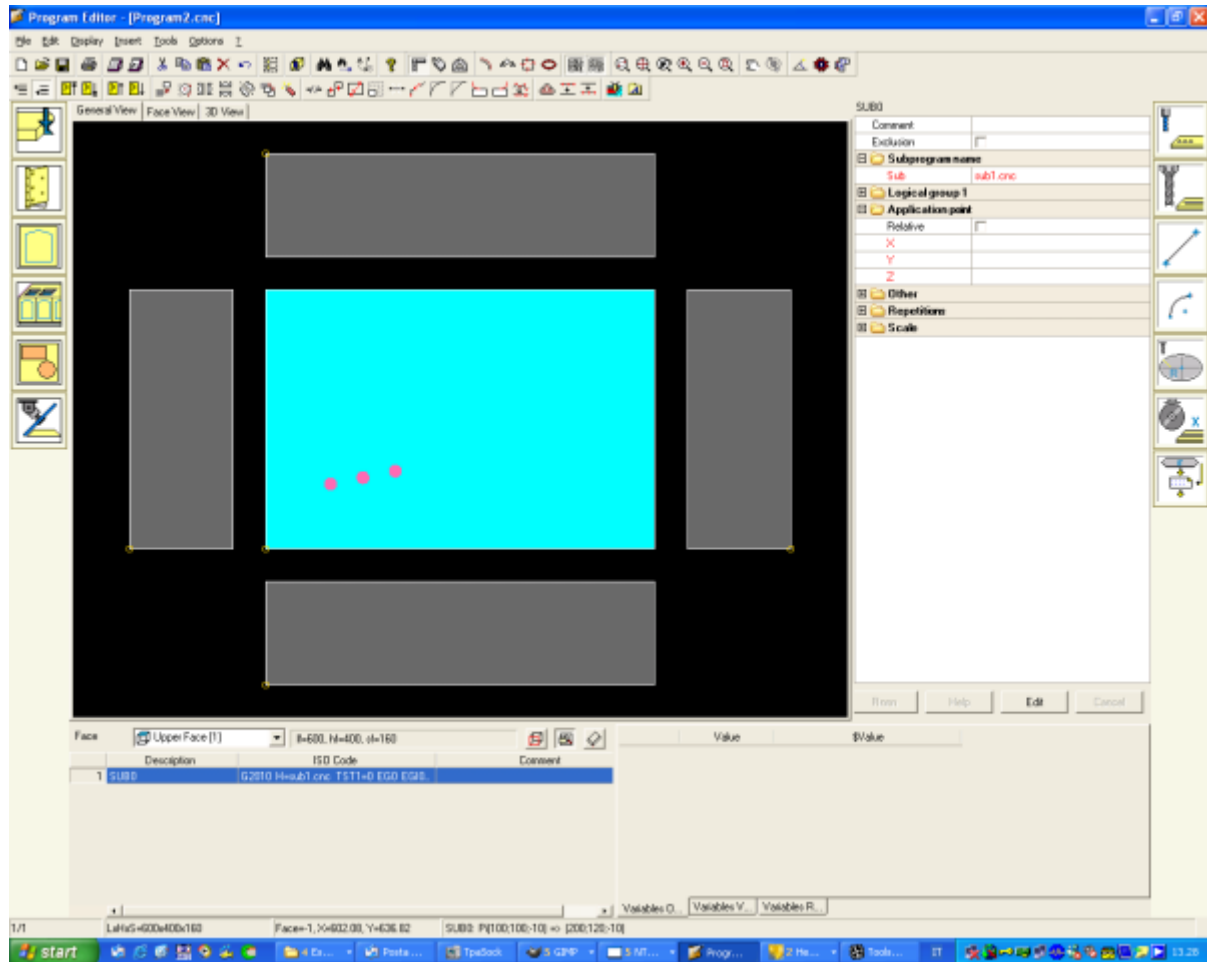


Fig: 8.2 - The SUB0 recalling function, with repeating

The Fig. 8.2 show the final result of a multiple recalling, on the face 1 of the main program, of a simple subroutine (named **sub1.cnc** and including a single Drill): the Repeating Factor (= 2) allows to add (to the basic recalling), other two, along a sloped straight line, characterized by the following two Cartesian component of inter-axis: Offset X = 50 mm and Offset Y = 20 mm.

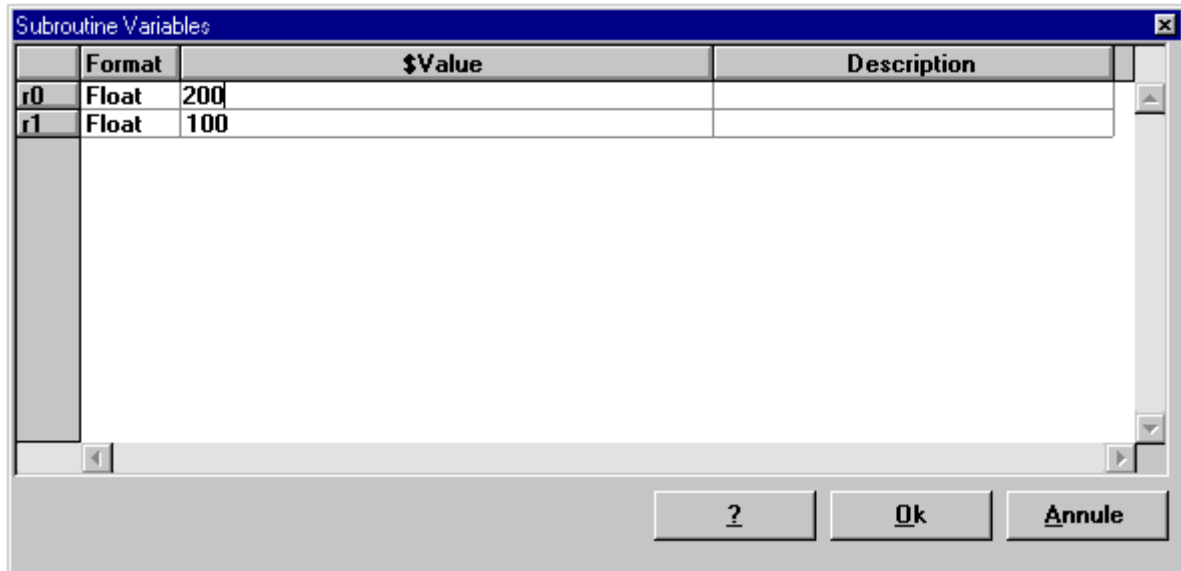
1.8.2 Subroutine Variables re-assigning

A very important feature, in the subroutine recalling mechanism, that allows the using possibilities increasing, is the possibility to modify its internal parameters, re-assigning them at the recalling moment.

This mechanism allows to obtain very different final results, according to the specific main program requirements, as, for instance, to "modulate" all the internal quotes to the current panel dimensions.

This possibility may be more conveniently used if Programmer took care, in the original Subroutine program, to indicate all the quotes (destined to be modulate by the Main) in terms of mathematical expressions based on its internal variables (**r0 – r299**): this logic may be applied to define, for instance, the most significant geometrical or technological characteristics (positions, inter-axes, diameters, depths, and so on) of the workings, when supposed to be redefined by the recalling program. In the subroutine creation, obviously the variables must be defined locally with a default values, to allow building and graphic representation, but marked with the special "re-assignable" ("wr") code.

At recalling, Programmer (using the "**Rnnn**" command-button) will must recall the corresponding "Variable Data Table" (see the Fig. 8.3) for the "current" parameters re-assigning: this last may be realized with a direct numerical assigning (as in the example) or using any mathematical expressions including Main program internal Variables, if initialized.



	Format	\$Value	Description
r0	Float	200	
r1	Float	100	

Figura 8.3 - Subroutine Variables re-assigning, at recalling

2 Workings Programming

2.1 Drilling workings

2.1.1 Piece Definition and dimensions

This is the first operation required for the piece characterization and will be supposed implicitly also in the following examples: it allows to configure the piece dimensions and some other auxiliary information, before to begin the Workings programming.

The characteristic "Piece parameters" recalling is performed *clicking*, with the mouse, the following Toolbar button:

Piece parameters



→Note : Thickness always refers to the Z+ axis direction: so the entry movement programming (hole depth, etc) implies the insertion of negative values.

→Note : The L, H, S variables can be used as parameters in the definition of the working application positions. So the programming of groups of similar programs becomes more easy and flexible.

2.1.2 DRILL (X,Y,Z) working

This Instruction allows to program a single Hole, on the selected face: in this example, we suppose the main face of a piece, which dimensions (1000x500x40) are defined in the previous exercise.

When the face has been selected (with the mouse) and visualized in the "Face View", the Drill working may be inserted *clicking* the following "Workings toolbar" button:

Drill (x,y,z)

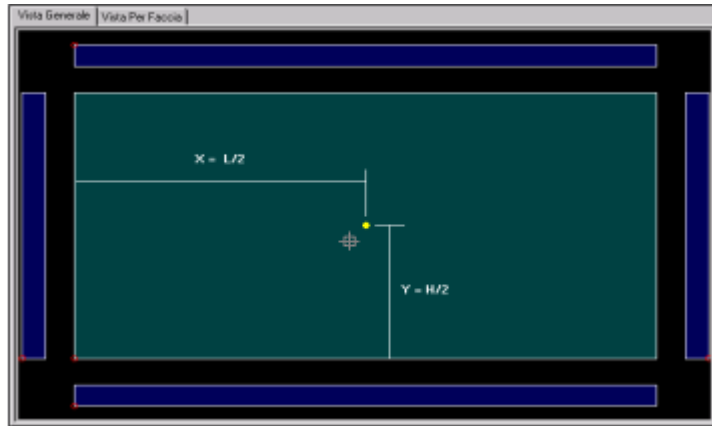


The corresponding Dialog box allows to enter:

- an informative comment,
- the Cartesian X and Y coordinates of the Hole, and the required Zp depth.

In the example, the Hole is programmed exactly in the panel medium; then, in alternative to the direct numerical programming (X = 500, Y = 250, Zp = -40) a "parametric mode" (X = L/2, Y = H/2, Zp = -40) may be more conveniently used, making reference to the panel current dimensions.

Drilling [X,Y,Z]	
Comment	
Exclusion	<input type="checkbox"/>
<input checked="" type="checkbox"/> Quotes	
X	l/2
Y	h/2
Zp	-40
Relative	<input type="checkbox"/>
<input checked="" type="checkbox"/> Other Quotes	
<input checked="" type="checkbox"/> Tools	
Diameter	
Tool	
Tool type	Blind bore drill



→ Note : The "Parametric Programming" allows the other general purpose variables using (more than Dimensions), as showed in the following.

The graphic window will display, at this point, the effective position of the programmed Hole.

→ Option: **Relative**

This option, not used in the example, allows to program the X/Y working quotes in "Incremental" mode, referred to the "current position", assumed as the immediately previous Working application quote.

This Option using will be described in the [Section 2.2](#).

→ Option : **Entry and Exiting Slowing quote**

During the entry movement of the tool in the piece, the downing Feed is normally considered fixed along the path. Nevertheless, overall in the case of "passing" drills, it's possible to program a slowing down in the initial part (Entry in the piece) or in the final (Exiting from the opposite side), in order to avoid a piece chipping!

Slowing down is defined, as percentage rate, in the "Technological Parameters" Session ("Working Feed").

In these two fields, the Entry and Exiting outlines length may be entered: the intermediate outline will be executed at the standard speed.

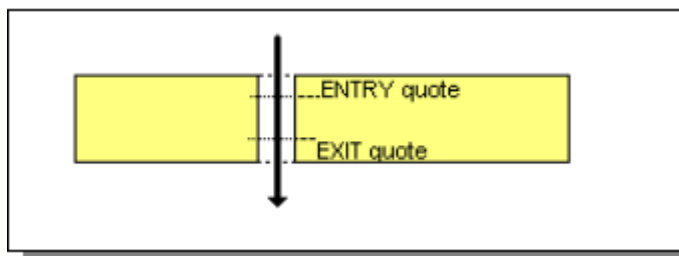


Fig. 1.1

Slowing down paths meaning.

The programmed quotes are referred to the initial (Entry) and final (Exit) paths. The internal path is executed to the nominal rate.

→ **Area: Tools**

This area allows to select the drilling Tool or directly (by the Group and Tool Number) or, implicitly, setting the "Diameter" and "Typology" attributes: in this last case, system will provide to find (if present) the correct tool for matching.

→ Area: Feed

The Feed values programming is not compulsory: the Tool selection implicitly may recall the corresponding "Default" values, as defined in the "Tool Parameters" session. Nevertheless, if not standard values are required, these fields allows the tool "Entry" Feed [m/min] and the Spindle "Rotation" Speed [rev/min] entering.

2.1.3 Drill Execution with the option "Relative"

In this example, "Relative" Option using is described: we suppose to insert a second Drill, after the previous (Ex. 1.1), programmed in incremental mode.

In this case, the X,Y quotes will be considered as an "incremental displacement" from the previous. Then, entering the values:

X = 50, Y = 25,

and clicking the Relative option

Drilling [X,Y,Z]	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
X	50
Y	25
Zp	-40
Relative	<input checked="" type="checkbox"/>
<input type="checkbox"/> Other Quotes	
<input type="checkbox"/> Tools	
Diameter	
Tool	
Tool type	Blind bore drill

the new drill will be positioned at the coordinates:

$$X = 500+50 = 550 \quad \text{and} \quad Y = 250+25 = 275$$

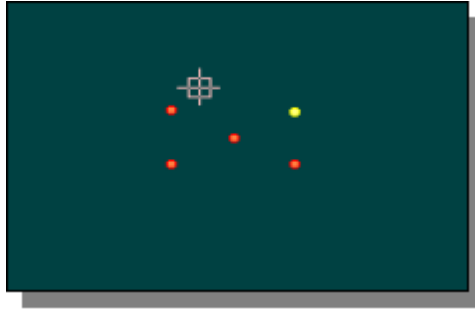
Now we suppose to require three other holes, positioned as to realize the four vertices of a rectangle of 100x50 dimensions

Solutions : there are two possibilities

A)... the first, to use an "Absolute" programming, then recalling more times the **DRILL (x.y.z)** instruction and entering the absolute coordinates [X=450, Y=275], [X=450, Y=225], [X=550, Y=225] and [X=550, Y=275], referred to the panel Origin.

B)...the second, to use the "Relative" programming, starting from the first absolute Drill:

First **Drill(x.y.z)** at the absolute coordinates X = 450, Y = 275: the following, with the "Relative" option, with the values: [X=0, Y=-50], [X=100, Y=0] and [X=0, Y=50].



2.1.4 DRILL in "Polar coordinates"

Allows to program a single Hole using a Polar Reference system: may be convenient, for instance, to define a set of holes along a circumference, when are to be disposed on different angles (otherwise is most convenient the special "Drill on Circle" macro-instruction, as described in the Ex. 1.6).

"Polar" programming requires to set the Pole (Origin of the reference), with the absolute Xcenter and Ycenter quotes, the distance of the Hole ("Module") and the Angle, referred conforming the trigonometric standard convention (positive if counterclockwise), relatively to the X+ direction.

In this example, we consider to insert 4 Drills, in correspondence to the four corners of a square, centered to the medium of the panel ($x = L/2$ and $y = H/2$, where the Pole will be defined) at a distance of 70mm.

We select the button:

DRILL (polar coordinates)



We enter, for the Pole, the values: Xcenter = L/2, Ycenter = H/2, supposing Zp = - 40 mm. The first Hole will be placed at the Angle = 45° and Module = 70 mm.

The following (with the same Module) will be characterized by the Angle values of : 135° (= 45°+90°), 225° and 315°.

Drilling (polar coordinates)	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
X Centre	l/2
Y Centre	h/2
Module	70
Angle	45
Zp	-40
Relative	<input type="checkbox"/>
<input checked="" type="checkbox"/> Other Quotes	
<input type="checkbox"/> Tools	
Group	
Tool	
Diameter	
Tool type	Blind bore drill

2.1.5 "Multi-tools Drilling"

Multi-tools Drilling



This instruction allows to program a multiple Drilling, by a direct Tool pattern selection.

In the example, two contiguous Tools (N. 1 and N. 2) are selected, in order to realize an Holes pair with a distance of 32 mm., as the selected tools pitch. The "Tools" field requires, in this case, a special syntax with the expression " 1 ; 2 ".

The Drill coordinates, absolute or relative, are to be referred to the first Tool of the list.

In the graphic displaying, all the programmed drills will be presented, according to the effective tools pattern.

Multiple Drilling	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
X	500
Y	250
Zp	-40
Relative	<input type="checkbox"/>
<input checked="" type="checkbox"/> Other Quotes	
<input type="checkbox"/> Tools	
Tool	1;2
<input checked="" type="checkbox"/> Speed	
<input checked="" type="checkbox"/> M Fields	

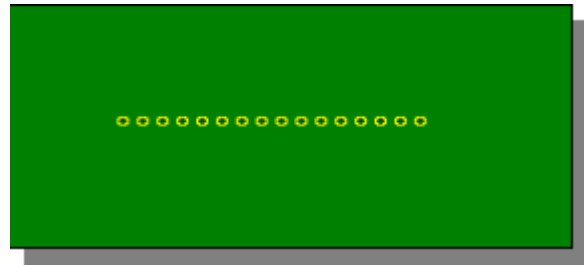
2.1.6 "Repeat X", "Repeat Y", "Repeat XY" macros

These special Macro instructions allows to program a set of Holes, disposed at constant distance, along the X direction ("Repeat X"), the Y direction ("Repeat Y") or along a generic sloped line ("Repeat XY"). The Holes sequence is programmed by the First and Last hole position and the inter-axis ("Pitch"), since system will compute automatically the total holes number.

- Repeat X



Repeat in X	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
Xi	l/2
Xf	l/2-300
Y	h/2
Zp	-40
Step	20
<input checked="" type="checkbox"/> Other Quotes	
<input type="checkbox"/> Tools	
Diameter	
Tool	
Tool type	Blind bore drill



This function requires: the X_i (initial), the X_f (final), and the Y quotes of the sequence, the Z_p depth and the distance between the holes (Pitch).

- **Repeat Y**

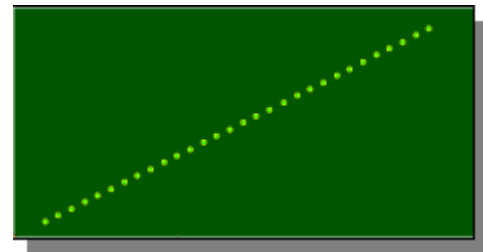


Similar to the previous, this function requires: the Y_i (initial), Y_f (final), the X quote of the sequence, the Z_p depth and the distance between the holes (Pitch).

- **Repeat XY**





Repeat in XY	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
Xi	900
Yi	450
Xf	50
Yf	25
Zp	-10
Step	32
<input checked="" type="checkbox"/> Other Quotes	
<input checked="" type="checkbox"/> Polar Progr.	
<input type="checkbox"/> Tools	
Diameter	
Tool	
Tool type	Blind bore drill



To define completely the sequence direction, this function requires: the Initial point (X_i, Y_i), the final point (X_f, Y_f), the Z_p depth and the Pitch.

In the example, the holes sequence starts from $X_i, Y_i = [900, 450]$ until to $X_f, Y_f = [50, 25]$ with a pitch of 50 mm.

2.1.7 "Fitting X" and "Fitting Y" Macros

- Fitting X 
- Fitting Y 

The "Fitting" operation is similar to the Repeat, since allows to program, easily, a set of Drills, at constant pitch, along the X or Y direction.

In this case, nevertheless, the drill inter-axis is not directly programmed but implicitly defined by the programmed tool pattern, that must include almost 2 tools.

The tool pattern is required to define a constant inter-axis between the selected spindles i.e. tools = 1;2;3 or tools 2;3;4. More, the selected tools sequence must be aligned compulsory or in the X direction (for **Fitting X**) or in Y direction (for **Fitting Y**).

Normally, with the typical Tools Head layout, only Pitches of 32mm or multiple may be realized.

2.1.8 "Drills on Circle" macro

This operation allows to program a sequence of Drill arranged along a circumference, with constant angular step.

In this Macro the following parameters must be entered :

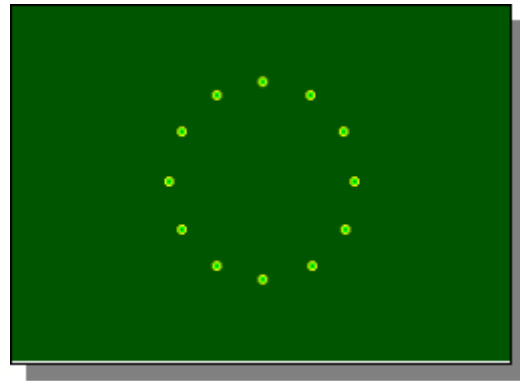
- The X,Y quotes of the Center of circumference
- the Number of required drills,
- the starting Angle (referred to the first drill, with the usual counterclockwise convention)
- the circumference Radius.

This macro is selected by the button:

- Drills on Circle 

In the example, a set of 12 drills (then with angular sector of 30 degrees) are programmed, on a circumference with Center X = 500, Y = 250 and radius 100.

Drilling on circle	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
X Centre	500
Y Centre	250
Zp	-10
Relative	<input type="checkbox"/>
Automatic distribution	<input checked="" type="checkbox"/>
Holes number	12
Angle	0
dA°	
Radius	100
<input type="checkbox"/> Other Quotes	



→ Option **dA°** :

In order to cancel this option select Automatic distribution and then OK, now the program put all the holes almost overlapped; we can observe that with a

zoom



of the piece.

Select again Edit and assign to dA° the value 20... then observe what happens !!!

The program distributed the points along the circumference with an amplitude of 20° between an arc defined by the point , and another arc.

2.1.9 "Drills on vertices of an inscribed Polygon" Macro

This macro-instruction, similar to the previous, allows to program a set of Drills on the vertices of a Polygon, inscribed in the programmed circumference. As in the previous case, all the Drills will be positioned over the circumference.

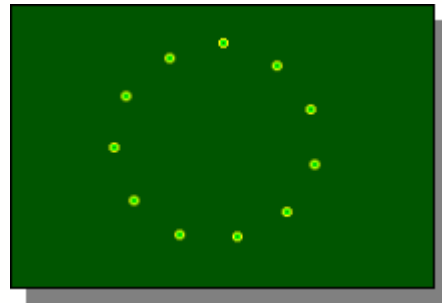
- Drills on vertices of an inscribed Polygon



The required parameters include:

- the Center of the circumference (i.e. X = 500, Y = 250),
- the drilling Depth (Zp=-10),
- the Number of Drills (Drills =11),
- the initial Angle (= 20 degrees) and
- the circumference Radius (=100).

Drilling on vertices of inscribed polygon	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
X Centre	500
Y Centre	250
Zp	-10
Relative	<input type="checkbox"/>
Holes number	11
Angle	20
Radius	100
<input type="checkbox"/> Other Quotes	



System will place eleven holes building a polygon with eleven sides, inscribed in the circumference.

2.1.10 "Drills on vertices of a circumscribed Polygon" Macro

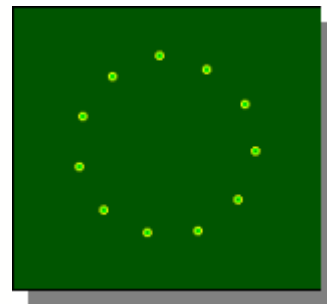
In this function, the Drills are positioned on the vertices of a circumscribed polygon, then externally of the circumference.

- Drills on vertices of a circumscribed polygon



The programming rules are the same as the previous case but the drill pattern will be outside the circumference.

Drilling on vertices of circumscribed polygon	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
X Centre	500
Y Centre	250
Zp	-10
Relative	<input type="checkbox"/>
Holes number	11
Angle	0
Radius	100



In the following example, we consider a circumscribed Square, with initial Angle = 45 degrees:

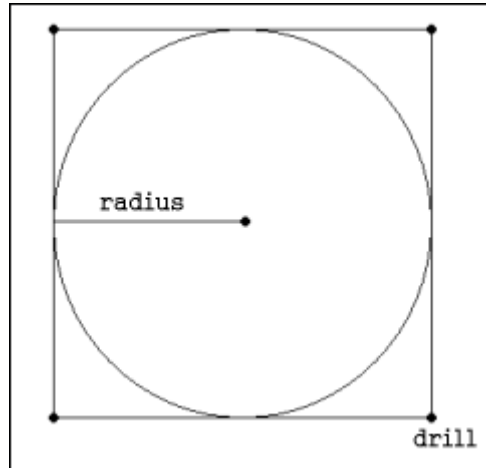


Fig. 1.8 : Drills pattern

Drills will be positioned outside the programmed circumference, since the square sides are tangent to this last.

In this case: Number = 4 and Angle = 45.

2.2 Sawing instructions

2.2.1 "Saw X", "Saw Y" and "Saw in degrees"

This group of functions allows to program "sawing" works, along the X direction ("Saw X"), Y direction ("Saw Y") or along a sloped line ("Saw in degrees")

The first is selected by the button:

Saw X



We consider to program a cut, with length = 300 mm., at the indicated quotes.

In the example, Sawing is set to the quote Y=250, starting from Xi = 300 and ending to Xf = 600, with 2 steps, the first with depth Zp= -10 and the second with Z2= -15.

Saw x	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
Xi	300
Xf	600
Y	250
Zp	-10
<input type="checkbox"/> Other Quotes	
Double pass	<input checked="" type="checkbox"/>
Z2	-15
Compute chord	<input type="checkbox"/>

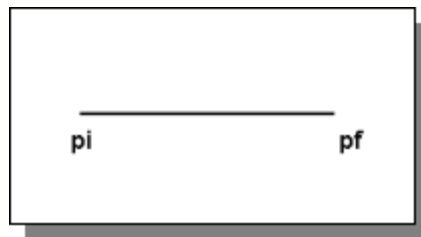
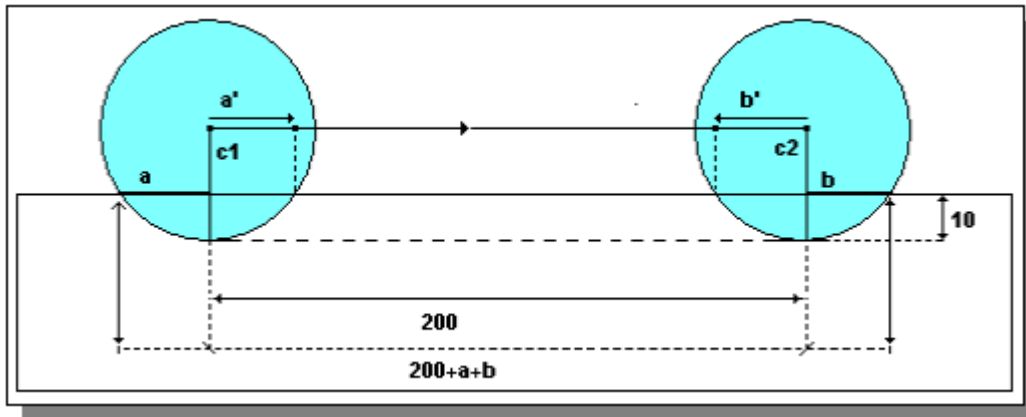


Fig.2.1a

Program executes sawing from Pi (300,250) to Pf (600,250)

→ Option: **Chord correction**



Programmer must consider that, in the sawing, the effective outline length will be greater than the programmed Pi-Pf segment, of the quantity, indicated in the figure as $(a+b)$, related to the Saw Radius and the final Depth.

The "Chord Correction" option allows to recall a special procedure to compute this "extra-quantity" and, consequently, to modify the Entry (Pi) and the Exit (Pf) points in order to obtain the required segment length.

→ Option: **Double Step**

Allows to program a second Step, enabling the Z2 field.

Sawing in Y direction is similar to the previous and is based on the same programming rules.

Now, we consider a **Sloped Sawing**, recalled by the button:

Saw degrees



In this case, the following parameters must be entered:

- Xi (=300) and Yi (=100) quotes of the Initial point
- The Inclination Angle (= 37°), with the usual trigonometric convention
- The Module (= 300), assumed as the total cutting length.

The other parameters (Depth, Options and so on) have the same meaning of the previous cases.

Saw oriented	
Comment	
Exclusion	<input type="checkbox"/>
☰ Quotes	
Xi	300
Yi	100
Alfa angle	37
Beta angle	
Module	300
Zp	-10
☰ Other Quotes	
Double pass	<input checked="" type="checkbox"/>
Z2	-15

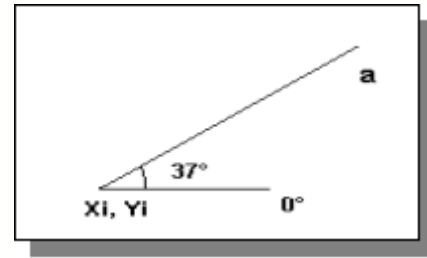


Fig.2.1b

Sawing starts from (Xi, Yi), with a length of 300 mm and inclination = 37° , as programmed in the Dialog box.

2.3 Mill Setup and Linear / Circular interpolations

2.3.1 "Mill Setup (x,y,z)" working

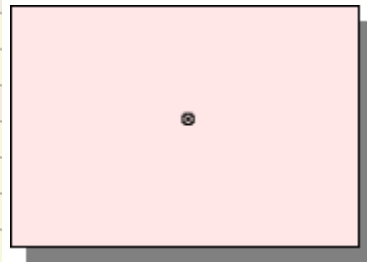
- **Mill Setup** :



This function allows to set the position of the Milling tool in a programmed point of the face, in order to prepare it for a milling profile, that will be geometrically defined in the following blocs.

Setup is characterized by the Cartesian coordinates of the Initial point (i.e. X = 500, Y = 250) and by the milling depth (may be changed during the following profile)

Mill Setup (X,Y,Z)	
Comment	
Exclusion	<input type="checkbox"/>
☰ Quotes	
Relative	<input type="checkbox"/>
X	500
Y	250
Zp	-10
☰ Other	
☰ Tools	
Tool 1	



- **Note** : Mill Setup must be always programmed at the beginning of every milling path.

In the following, the different geometric elements, that may be included in the profile, will be described.

Every geometric element will assume, as **Initial** point, the **final** point (indicated as "Current point") of the **immediately previous bloc**.

→ Option: **Relative**

As usually, enables the "incremental" programming, assuming the programmed quotes values in incremental mode, referred to the Application point of the previous Working.

→ Option: **Point Link**

May be very convenient, in programming, to prepare a pre-defined milling path, defining them as "stand alone" but thinking to recall it recurrently, more times, as to be inserted into other profiles. In this case, this profile may be programmed alone and stored as a Subroutine, but normally it will be recalled by other programs ("Main") as to be linked to the profile in building, hooking them in correspondence with the "current" point (the final point of the previous main program bloc, before recalling).

Then the "Point Hooking" option supplies this mechanism, linking the recalled profile, then erasing its internal Setup and moving them as to place its initial point in correspondence of the Current.

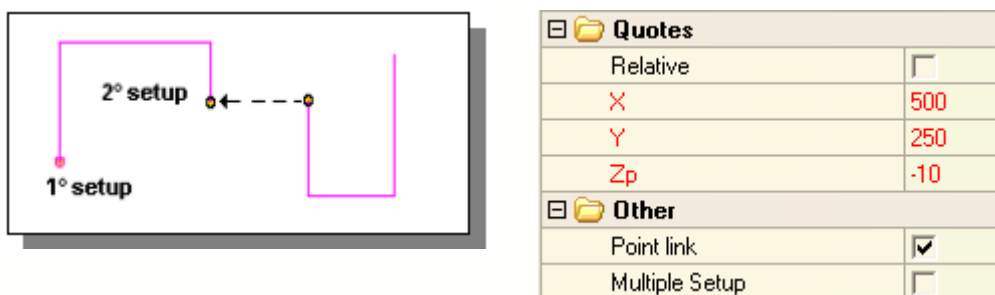


Fig. 3.1a: In the figure, the first Mill Setup is indicated: at the end of the third line (2° setup) an external "subroutine" path may be linked with the "Point Link" option.

→ Option: **Multiple Setup**

In programming may occurs that a defined profile must be worked, in sequence, by more different tools, for instance for "Rough shaping", "Finishing" and "Post forming", without changing in the geometrical path.

The "Multiple Setup" option allows to avoid a repetitive geometrical definition of the profile, allowing to define it only one time, and assigning to it a sequence of different Tools Setup.

The first programmed Mill Setup will indicate the first working tool, with the usual set of parameters and with this Option enabled. All the other Setup will follow immediately, according to the working sequence: all of these must be characterized by the same X,Y "Application" quotes: in effect, it will be sufficient to leave empty the corresponding fields and enable the "Point Hooking" option or, alternatively, to enter Zero quotes and select the "Relative" option. In any case, all the Setup instructions must be characterized by the "Multiple Setup" option enabled!

After the last Tool Setup, the geometrical elements will follow as usually.

Mill Setup (X,Y,Z)		Mill Setup (X,Y,Z)		Mill Setup (X,Y,Z)	
Comment		Comment		Comment	
Exclusion	<input type="checkbox"/>	Exclusion	<input type="checkbox"/>	Exclusion	<input type="checkbox"/>
Quotes		Quotes		Quotes	
Relative	<input type="checkbox"/>	Relative	<input type="checkbox"/>	Relative	<input checked="" type="checkbox"/>
X	100	X		X	
Y	100	Y		Y	
Zp	-10	Zp		Zp	
Other		Other		Other	
Point link	<input type="checkbox"/>	Point link	<input checked="" type="checkbox"/>	Point link	<input type="checkbox"/>
Multiple Setup	<input checked="" type="checkbox"/>	Multiple Setup	<input checked="" type="checkbox"/>	Multiple Setup	<input checked="" type="checkbox"/>

Fig. 3.1b: In the first figure, the first Setup is programmed with the Multiple Setup option enabled.

In the following two, we can see two different programming modes for the second Setup:

- in the second Table, the Point Link option enabled
- in the third, a nul quotes are entered, with the Relative option enabled.

Concerning the "Technological area" programming and the Tool selection, Programmer is suggested to make reference to the [Sect. 1.5](#), of the present Manual.

2.3.2 "Mill Setup" (polar coordinates)

- Mill Setup(polar coordinates) :



Similar to the [Mill Setup \(x,y,z\)](#), this instruction allows to program the Setup point with a Polar Reference System using, as already described in the Ex. 1.3 (**Drill (polar coordinates)**)

In the table, the following parameters are required:

- the Xcenter and Ycenter quotes, referred to the Pole position
- the Inclination Angle and the vector Module
- the Zp depth

In the present example, Setup is positioned at the coordinates:

$$X = X_{center} + Module * \cos 60^\circ = 500 + 100 * 0.5 = 550$$

$$Y = Y_{center} + Module * \sin 60^\circ = 250 + 100 * 0.866 = 336.6..$$

Mill Setup (polar coordinates)	
Comment	
Exclusion	<input type="checkbox"/>
Quotes	
Relative	<input type="checkbox"/>
Zp	
Other Quotes	
X Centre	500
Y Centre	250
Angle	60
Module	100

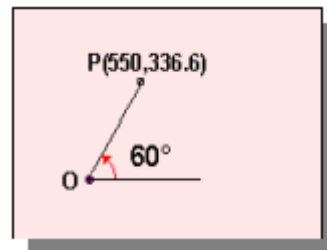


Fig. 3.2 :

Program defines the Setup point in "polar" mode, with the "P" point characterized by a Module =100 mm, Angle = 60° from a Pole "O" = (500,250)

2.3.3 "Oriented Mill" Setup

- Oriented Mill Setup :



This function allows to define the initial position, in the space, of a Rotating Tool, using the primary rotation Alfa axis (referred to the vertical axis) and the secondary branding Beta axis (placed on the XZ plane when Alfa=0).

The Setup point may be programmed in Cartesian or Polar modes, using the following two different macro-instructions:

Oriented-tool Mill Setup	
Comment	
Exclusion	<input type="checkbox"/>
☐ Quotes	
Relative	<input type="checkbox"/>
X	
Y	
Zp	
☐ Other Quotes	
Alfa angle	0
Beta angle	0
☐ Other	
☐ Tools	
Tool 1	
☐ Tool radius	
Tool radius compensation	None
Tool radius value	
Progressive Tool Radius In	<input type="checkbox"/>
Progressive Tool Radius Out	<input type="checkbox"/>
☐ Speed	
Entry speed	
Speed on inserted fillets	
Spindle RPM's	
Rotation	0 - Default
☐ M Fields	
M1	
M2	
M3	

a) Cartesian Programming

The Setup point is defined by the X and Y quotes (in Absolute or Relative mode), referred to the selected working face. Zp indicates the initial Depth.

Note: For a detailed description about the sloped faces programming rules, Reader is suggested to make reference to the Sect. [2.3.25](#)

Oriented-tool Mill Setup(polar coordinates)	
Comment	
Exclusion	<input type="checkbox"/>
☐ Quotes	
Relative	<input type="checkbox"/>
Zp	
☐ Other Quotes	
Alfa angle	0
Beta angle	0
X Centre	
Y Centre	
Angle	
Module	
☐ Other	
☐ Tools	
Tool 1	
☐ Tool radius	
Tool radius compensation	None
Tool radius value	
Progressive Tool Radius In	<input type="checkbox"/>
Progressive Tool Radius Out	<input type="checkbox"/>
☐ Speed	
Entry speed	
Speed on inserted fillets	
Spindle RPM's	
Rotation	0 - Default
☐ M Fields	
M1	
M2	
M3	


b) Polar Programming

The Setup point is defined by the Pole position (XCenter, YCenter), the Module and the Angle. Zp assumes the usual depth meaning.

2.3.4 Linear milling - "L01 (X,Y,Z)"

This function allows to program a Linear milling, starting from the current point, until the programmed final point.

We suppose a Mill Setup programmed at the quotes $X = 100$, $Y = 100$, $Z_p = -10$, and we select:

- **L01 (X,Y,Z)** 

In this example, we enter the Final point quotes (absolute) $X_f = 500$, $Y_f = 100$, $Z_p = -10$:

Program will display the mill path, until the programmed point.

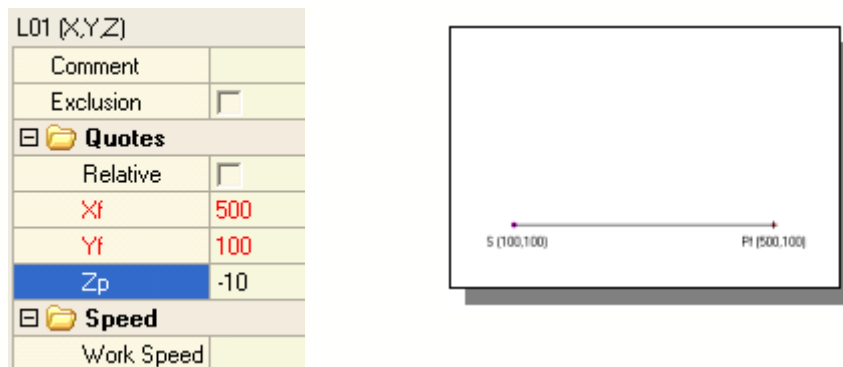


Fig. 3.4a : Linear milling on the face 1.

For demonstrative purpose, Programmer is suggested to build a rectangle, with dimensions 400×200 , using the relative modality, until to obtain the result showed in the **Fig. 3.4b**

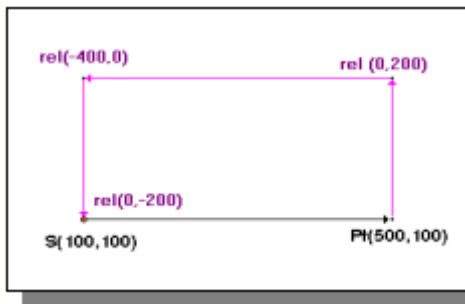


Fig. 3.4b : this rectangle has been built with five steps: the Mill Setup, a first Linear in absolute coordinates followed by three Linear in relative programming. Programmer must remember that, in relative mode, the reference point is the final point of the previous bloc.

Solution : the relative coordinates of the following Linear bloc will be:

- first milling: $X_f = 0$, $Y_f = 200$,
- second milling: $X_f = -400$, $Y_f = 0$,
- third milling: $X_f = 0$, $Y_f = -200$.

→ Field: **Working Feed**

It allows, if required, to modify the working Feed for this bloc, changing the original value defined in the Setup or in the previous blocs.

2.3.5 Linear milling - "L02 (pole,A,U)"

In this function, similar to the "[L01\(X,Y,Z\)](#)", the Final point is programmed with a Polar Reference system, then with the usual rules including the Pole, Angle and Module defining.

- **L02 (pole,A,U)**



The following parameters are required:

- Pole coordinates : X center = 300, Y center = 100
- Milling Depth : Zp = -10 (optional, if equal to the previous bloc depth)
- Module = 150,
- Angle = 135.

- Note : the Pole coordinates may be programmed in Absolute or Relative mode:

L02 (pole,A,U)	
Comment	
Exclusion	<input type="checkbox"/>
<div style="border: 1px solid black; padding: 2px;"> Q Quotes </div>	
Relative	<input type="checkbox"/>
X Centre	300
Y Centre	100
Zp	-10
Angle	135
Module	150
<div style="border: 1px solid black; padding: 2px;"> S Speed </div>	

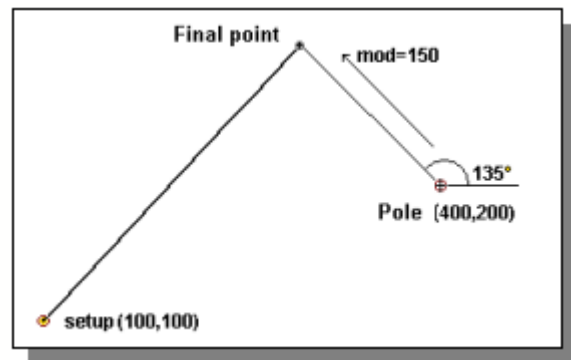


Fig. 3.5a :

In the figure, the final point of the segment is programmed with Polar reference and the following parameters:

- Center (Pole) = (400,200),
- Module = 150
- Angle = 135°

For demonstrative purpose, we can see what happens if we change, maintaining the Pole in a fixed position, the Module or the Angle of the Polar reference.

Changing the Angle, the Final point will move according the **Fig. 3.5b**

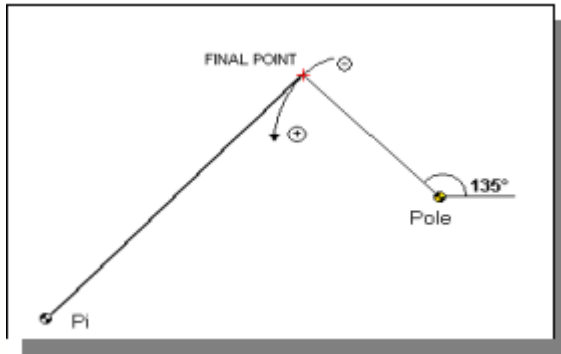


Fig. 3.5b

Changing the Angle, the Final point will move along the indicated arc of circumference, centered to the Pole

If the Module is changed, the Final point moves according to the **Fig. 3.5.c**

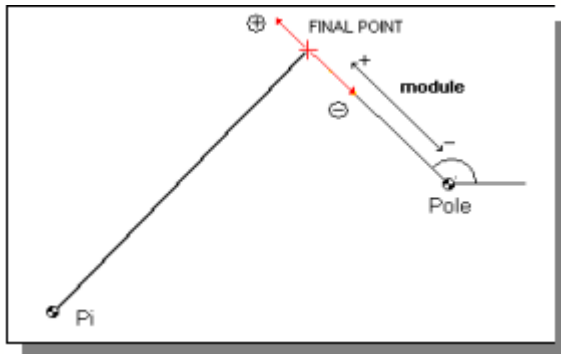


Fig. 3.5c:

Changing the Module, the Final Point will move along the vector defined by the angle.

2.3.6 Linear milling - "L03 (A,U)"

L03 (A,U)



This function is similar to the "[L02](#)", but implicitly assumes the Pole on the "current" point, then on starting point of the segment.

Then, in this case, the Pole coordinates are not required.

L03 (A,U)	
Comment	
Exclusion	<input type="checkbox"/>
<div style="display: flex; align-items: center;"> <div style="margin-right: 5px;">☞</div> <div>Quotes</div> </div>	
Relative	<input type="checkbox"/>
Zp	-10
<div style="display: flex; align-items: center;"> <div style="margin-right: 5px;">☞</div> <div>Other Quotes</div> </div>	
Angle	45
Module	150
<div style="display: flex; align-items: center;"> <div style="margin-right: 5px;">☞</div> <div>Speed</div> </div>	

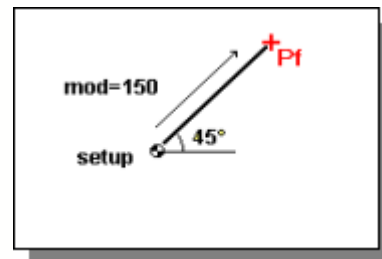


Fig. 3.6

The milling segment is defined by the Module (=150mm) and by the Angle (=45).

2.3.7 Linear milling - "L04 (pole,A,X)"

Similar to the L02, in this case the Final point is defined by the Pole, the Angle and by the effective Xf coordinate.

- L04 (pole,A,X)

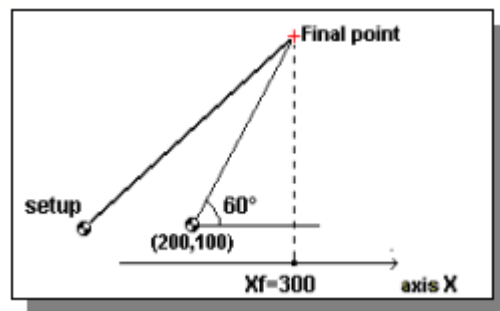


In the example, the following values have been programmed:

- X center = 200, Y center = 100,
- Zp = -10,
- Xf = 300, angle = 60°.

The effective position of the Final point will be : Xf = 300 (programmed) and Yf = 273.20

L04 (pole,A,X)	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
Relative	<input type="checkbox"/>
X Centre	200
Y Centre	100
Zp	-10
Xf	300
<input type="checkbox"/> Other Quotes	
Angle	60
<input type="checkbox"/> Speed	



Note: Similar to the present, the " L05 (Pole, A, Y)" function requires, for the Final point definition, the effective Yf quote.

2.3.8 Linear millings - "L06 (A,X)" and "L07 (A,Y) "

They are similar to the "L04", for the Final point definition, but assume implicitly the Pole coincident to the "current" point (Starting point of the segment).

In the example, we recall the function:

- L06(A,X)



with the following values: Zp = -10, Xf = 250, Angle = 60

L06 (A,X)	
Comment	
Exclusion	<input type="checkbox"/>
☰ Quotes	
Relative	<input type="checkbox"/>
Zp	-10
Xf	250
☰ Other Quotes	
Angle	60
☰ Speed	

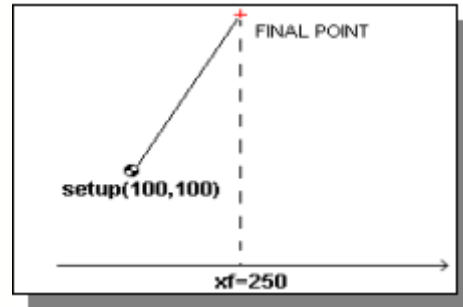


Fig. 3.8 :

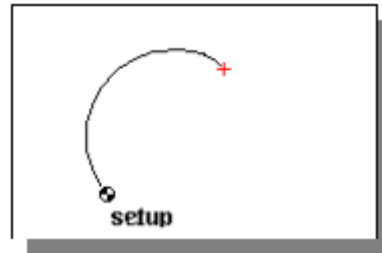
The Final point will be computed at the quotes :
 $X_f = 250$ (programmed) and $Y_f = 359.80$ (computed).

2.3.9 Linear milling - "L08 (Z,tangent)"

This function allows to program a segment, with programmed length, assuming implicitly, as direction, the tangent exiting from the previous bloc.

In the example, we suppose has been programmed a Mill Setup (at the coordinates $X=100$, $Y=100$ and $Z_p=-10$) and, after, a Arc "A04 (Arc by 3 points)" function, as showed in the figure:

A04 (three points) on plain XY	
Comment	
Exclusion	<input type="checkbox"/>
☰ Quotes	
Relative	<input type="checkbox"/>
Xf	200
Yf	200
Zp	-10
X1	110
Y1	200
Arc rot. direction	Automatic
☰ Speed	



Selecting

- **L08 (Z,tangent)**



we must enter the only "Module" parameter, then the segment required length.

L08 (Z.tangent)	
Comment	
Exclusion	<input type="checkbox"/>
📁 Quotes	
Relative	<input type="checkbox"/>
Zp	-10
📁 Other Quotes	
Module	100
📁 Speed	

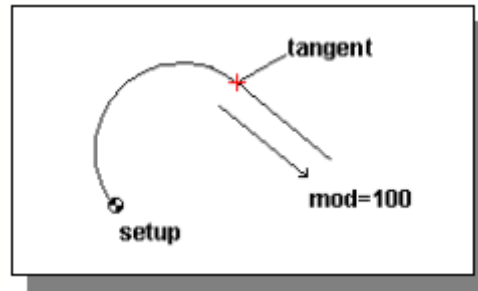


Fig. 3.9

Program generates the new segment with Tangent continuity respect to the previous arc, with the programmed length.

→ **Note:** The Zp field allows to modify Depth, on the final point, respect to the starting value. Then the Tangent continuity must be considered only related to the XY plane projection of the segment.

→ **Note:** Similar to the present, the "**Linear L09 (module)**" function do not admit a change of Depth; consequently the Zp field is missing.

2.3.10 Linear milling - "L10 (closure)"

This function allows to close a profile, forcing a linear segment from the current point to the Setup point

No geometrical parameter is required (eventually only the Feed).

- **L10 (closure)** :



The working Data Table includes :

L10 (closure)	
Comment	
Exclusion	<input type="checkbox"/>
📁 Speed	
Work Speed	

2.3.11 Linear milling - "L12 (Tginiz (P1,P2 o A), U, Z)"

This function is similar to the "**L08**", since allows to program a linear milling defining the tangent: in this case, nevertheless, this last is not assumed from the previous bloc, but is expressly programmed, or directly, by the "A" Angle or, implicitly, by a point pair [P1 and P2], for a direction definition.

→ **Note:** This programming mode, for the Tangent definition (A or P1-P2) will be also used in the following, in the set of "Arc" functions.

In the example, we select the function:

- **L12 (Tginiz (P1,P2 o A), U, Z)**



where we can see the different Tangent programming mode:

- In the first case, Tangent is defined by the point pair P1 (100,100) and (300,300), forcing an inclination of 45°.
- In the second, these fields are ignored, with a direct "Angle" programming (= 45).

The final result is the same: when direction is defined, the milling segment is completely characterized by its length (=Module).

L12 (Tangent P1,P2 or A,U,Zf)	
Comment	
Exclusion	<input type="checkbox"/>
☰ Quotes	
Relative	<input type="checkbox"/>
Zp	
☰ Other Quotes	
X1	
Y1	
X2	
Y2	
Angle	
Module	

Fig. 3.11a : Example of the L12 function Table

L12 (Tangent P1,P2 or A,U,Zf)	
Comment	
Exclusion	<input type="checkbox"/>
☰ Quotes	
Relative	<input checked="" type="checkbox"/>
Zp	0
☰ Other Quotes	
X1	100
Y1	100
X2	300
Y2	300
Angle	
Module	300
☰ Speed	

Fig. 3.11b : In this case, the segment direction is defined by the points pair P1-P2

L12 (Tangent P1,P2 or A,U,Zf)	
Comment	
Exclusion	<input type="checkbox"/>
☰ Quotes	
Relative	<input checked="" type="checkbox"/>
Zp	0
☰ Other Quotes	
X1	
Y1	
X2	
Y2	
Angle	45
Module	100
☰ Speed	

Fig. 3.11c : In this case, the segment direction is defined by direct Angle entering.

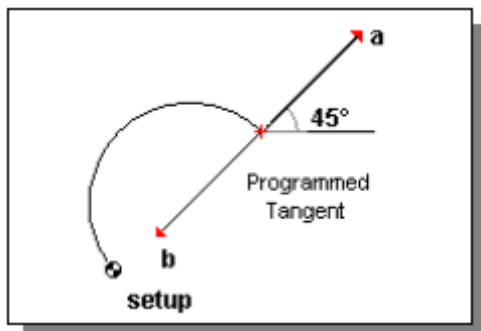


Fig. 3.11d : Programmer must take in account the correct P1 – P2 sequence.

If the two points are reversed, system will assume an opposite direction for the segment, then executing the segment "b", opposite to the original "a" segment (with +45°)

2.3.12 Arc "A01 (center) on plane XY"

This function allows to define a circular arc to connect the current point (initial point) to a Final programmed point. The A01 Function defines the arc with its Center direct assigning.

In the example, we suppose that the Mill Setup (x,y,z) has been programmed in the medium of the panel (→ Ex. 3.0). Then selecting:

- A01 (center) on plane XY



the following parameters must be entered:

- the Xf and Yf values for the Final point of the arc (in Absolute or in Relative).
- the Zp depth,
- the coordinates of the Center (Xcenter, Ycenter), always assumed in incremental mode respect to initial point (current point)
- the moving sense (Clockwise / Counterclockwise) of the arc

A01 (center) on plain XY	
Comment	
Exclusion	<input type="checkbox"/>
<input checked="" type="checkbox"/> Quotes	
Relative	<input type="checkbox"/>
Xf	100
Yf	0
Zp	-10
<input checked="" type="checkbox"/> Other Quotes	
X Centre	50
Y Centre	0
CCW	<input type="checkbox"/>
<input checked="" type="checkbox"/> Speed	

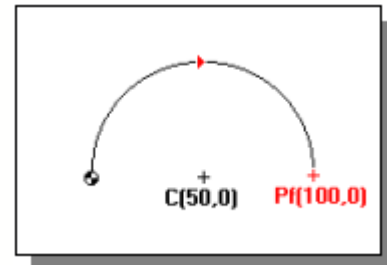


Fig. 3.12 : 1) Mill Setup: (l/2,h/2);
2) Final point Xf, Yf (600,250)
3) Center of the arc: (550,250)

Taking in account of the position of the initial point, in this case assumed on the Setup ($X_1=500$, $Y_1=250$), program realizes an Arc with Radius = 50 mm and Center on the point ($X_c=500+50=550$ and $Y_c=250+0=250$).

The Final point coordinates, programmed in Relative, will result $X_f=500+100=600$ and $Y_f=250+0=250$.

→ **Note:** This programming mode, assigning all the four values (2 for the Center and 2 for the Final point) will result, in general, "hyperstatic", in the sense that incorrect values may cause situations with geometrical incompatibility, then causing "Programming Error" messages.

→ **Note:** the A01 function indicates an Arc on the plane XY. Similar functions (the arc **A05** and the **A06**) define, with similar rules, a circular arc in the other XZ and YZ planes.

- **A05 (center) plane XZ**



- **A06 (center) plane YZ**



2.3.13 Arc "A04 (three points) plane XY"

In this function the Arc is defined by its Initial (implicit), Final and by an Intermediate point, must not be aligned to the other two.

As usually, the initial point is assumed on the current point.

As example, we consider as current point the Final point of the previous instruction. We select:

- A04 (three points) in XY



Arc generated with the following values (Initial point :X=600, Y=250):

- Final point (in Relative): Xf=100, Yf=0 (then, in absolute X=600+100=700, Y=250+0=250)
- Internal Point (in Relative): X1=50, Y1=100 (then, in absolute X=600+50=650, Y=250+100=350)

A04 (three points) on plain XY	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
Relative	<input type="checkbox"/>
Xf	100
Yf	0
Zp	-10
X1	50
Y1	100
Arc rot. direction	Automatic
<input type="checkbox"/> Speed	

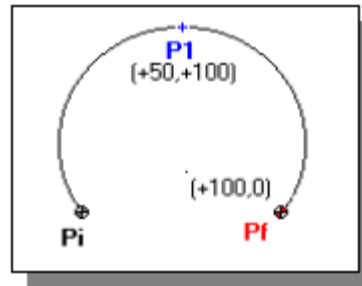


Fig. 3.13 :
 Initial point = Pi (600,250);
 Final point = Pf (700,250);
 Third point = P1 (650,350);

→ Note: When the intermediate point position is defined, system computes automatically the moving sense on the arc, as to include this programmed point.

→ Note: Similar to the A04 function, the following two allows to realize the same operation on the other two planes (XZ and YZ).

- A07 (3 points) plane XZ



- A08 (3 points) plane YZ



2.3.14 "A11 (radius, smaller arc) on plane XY"

With this function, the Arc is defined by its Initial point (implicit), the Final point and by the Radius: between the two possible geometrical solutions, automatically is assumed the corresponding to the smaller length Arc (viewed by the Center under an angle less than 180 degrees).

- A11 (radius, smaller arc) on plane XY



we suppose the initial point on the Setup (500,250). Entering:

- Xf = +100,
- Yf = +10 (in Relative)
- Depth Zp=-10
- Radius = 70

we obtain the following result:

A11 (radius, smaller arc) on plain XY	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
Relative	<input checked="" type="checkbox"/>
Xf	100
Yf	10
Zp	-10
Radius	70
CCW	<input type="checkbox"/>
<input type="checkbox"/> Speed	

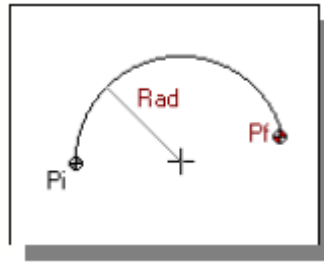


Fig. 3.14a :

- 1) Initial point P1= 500,250
- 2) Final point P2=600,260
- 3) Radius of the arc = 70

→ **Note:** The Clockwise/Counterclockwise sense of the arc may be also available. The Center choice (between the two possible) is, on the contrary, forced when the "smaller length" condition has been selected.

→ **Note:** Similar to the present, the function "**A12 (radius, greater arc) on plane XY**" allows to select the greater length arc.

- **A12 (radius, greater arc) on plane XY**



A12 (radius, greater arc) on plain XY	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
Relative	<input checked="" type="checkbox"/>
Xf	100
Yf	10
Zp	-10
Radius	70
CCW	<input type="checkbox"/>
<input type="checkbox"/> Speed	

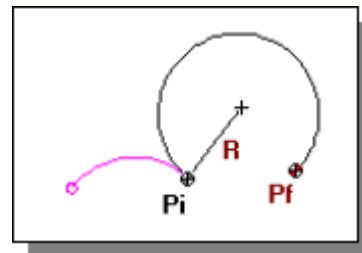


Fig. 3.14b :

Arc is built with the same Radius and Final point of the previous case: the greater Arc option causes a different Center choice.

→ **Note:** As showed, a total of four different arcs may connect the points 1 and 2, with imposed radius: 2 clockwise (Greater and Smaller) and other two counterclockwise.

2.3.15 Arc "A13 (center and final angle) on plane XY"

In this function Arc is defined by its Initial point, the Center (programmed) and by the Final point, implicitly determined by the viewing Angle (respect to the Center).

A13 (center and final angle) on plane XY



Programming requires then the Zp depth, the Center coordinates, the viewing Angle and the moving sense (CW or CCW).

- **Note :** The Xcenter, Ycenter quotes are always considered as incremental respect to the current point: the "Relative" option is referred only to the Zp parameter.

In the example, the following values are entered:

- X center = +100,
- Y center = +100
- Angle = 90,
- Sense: Clockwise

A13 (center and final angle) on plain XY	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
Relative	<input type="checkbox"/>
Zp	
<input type="checkbox"/> Other Quotes	
X Centre	100
Y Centre	100
Angle	90
CCW	<input type="checkbox"/>
<input type="checkbox"/> Speed	

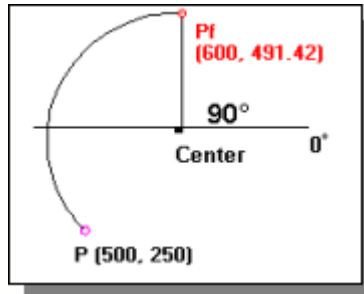


Fig. 3.15 :
Program introduces an Arc, with Center (600,350), until to the Final point [X=600, Y=491.42].

→ Note: System computes automatically the effective coordinates of the Center ($X_c = 500 + 100 = 600$, $Y_c = 250 + 100 = 350$), the Radius ($R = 141.42$), then obtaining the Final point coordinates:

$$X_f = X_c + R \cdot \cos(90) = 600 + 0 = 600$$

$$Y_f = Y_c + R \cdot \sin(90) = 350 + 141.42 = 491.42$$

2.3.16 Arc "A15 (entry tangent) on plane XY"

This function allows to realize an Arc only programming the Final point, since assumes, by default, that the Entry tangent is equal the tangent exiting from the previous bloc (arc or segment).

We select:

- **A15 (entry tangent) on plane XY**



and we suppose that the previous bloc should be an horizontal segment, with Final point (Point Pi) at the quotes: 200,200. The Entry tangent, forced by the previous bloc, will assume Angle = 0.

In the example, with the entered values, the Final point is computed at the quotes:

$$X_f = X_i + 100 = 200 + 100 = 300$$

$$Y_f = Y_i + 100 = 200 + 100 = 300$$

A15 (tangent in entry) on plain XY	
Comment	
Exclusion	<input type="checkbox"/>
<input checked="" type="checkbox"/> Quotes	
Relative	<input checked="" type="checkbox"/>
Xf	100
Yf	100
Zp	-10
Arc rot. direction	Automatic
<input checked="" type="checkbox"/> Speed	

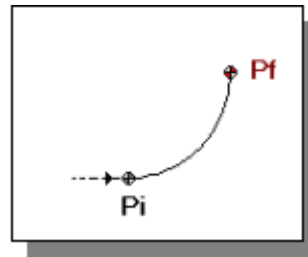


Fig.3.16 :

- 1) Initial point Pi supposed at the quotes (200,200)
- 2) Final point Pf computed at the coordinates (300,300)

2.3.17 Arc "A16 (initial tangent, final point) on plane XY"

This function is similar to the "A15": nevertheless, in this case, the initial tangent of the arc is not forced by the previous bloc but directly programmed, as usually, with the two alternative modalities:

- 1) with the points pair P1(x1,y2) and P2(x2,y2), defining an oriented segment, or
- 2) by the Angle.

→ Note: The P1 and P2 points may be any and, in particular, the P1 is not compulsory coincident with the Initial point of the Arc.

Example: we select:

- **A16 (initial tangent, final point) on plane XY**



entering the following values:

- for the Final point: Xf=300, Yf=150 (programmed in Absolute)
- defining the P1 (X1,Y1) and P2 (X2,Y2) points, forcing, in this case, a direction = 45°.

A16 (tangent in entry, final point) on plain XY	
Comment	
Exclusion	<input type="checkbox"/>
<input checked="" type="checkbox"/> Quotes	
Relative	<input type="checkbox"/>
Xf	300
Yf	150
Zp	-10
Arc rot. direction	Automatic
<input checked="" type="checkbox"/> Initial Angle	
X1	180
Y1	180
X2	200
Y2	200
Initial Angle	
<input checked="" type="checkbox"/> Speed	

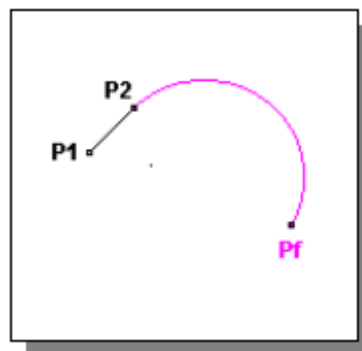


Fig. 3.17 : Arc begins (in the figure the initial point coincides with the point 2, but this is not compulsory) with Initial tangent defined by the P1-P2 segment and ends on the programmed Final point (point Pf).

2.3.18 Arc "A17 (exit tangent) on plane XY"

This Function allows the Arc building with opposite logic of the "A16": Arc begins, as usually, from the current point, and is characterized by a programmed Final point (Xf, Yf) and by the Exit Tangent, where direction is programmed by the P3(x3,y3) and P4(x4,y4) pair or, directly, by the final Angle.

- A17 (exit tangent) on plane XY



Now, entering the following coordinates:

- final point: Xf = 350, Yf = 300,
- P3 point : X3 = 350, Y3 = 300, (coincident with Pf, but not compulsory)
- P4 point : X4 = 400, Y4 = 400

A17 (tangent in exit) on plain XY	
Comment	
Exclusion	<input type="checkbox"/>
<div style="border: 1px solid black; padding: 2px;"> Q Quotes </div>	
Relative	<input type="checkbox"/>
Xf	350
Yf	300
Zp	-10
X3	350
Y3	300
X4	400
Y4	400
Final Angle	
Arc rot. direction	Automatic
<div style="border: 1px solid black; padding: 2px;"> S Speed </div>	

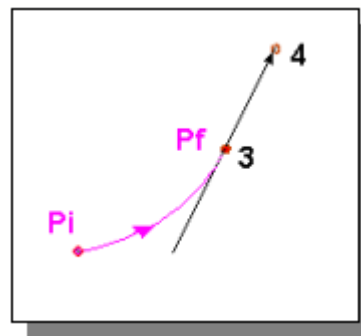


Fig. 3.18 :
Exit direction is forced by the P3(350,300)-P4(400,400) segment.

2.3.19 Double Arc "Arc1-Arc2 (C1,C2)"

The "Double Arc" functions allows, as showed also in the following examples, to program, in very easy mode, two consecutive arcs with continuity (or opposite) of tangent in the intersection point. The second arc may be programmed with the same or opposite sense (CW or CCW).

Normally, conforming to the programmed parameters, system try to impose the tangent continuity or, if impossible, to select an opposite direction (180 degrees).

If inconsistent data are entered, a "Programming Error" message will be generated.

We consider the first function "Double Arc by Centers":

- Arc1-Arc2 (C1,C2)



The following parameters are required :

- the C1 center of the first arc, [X center, Y center].
- the final point of the second arc, [Xf, Yf].
- the C2 center of the second arc (assumed in relative respect to the final point), [X center, Y center].

We suppose a Mill Setup positioned at the coordinates (100,100,0), then we enter the following

values:

- Quote arc1 : X center = 50, Y center = 0;
- Quote arc2 : X center = -50, Y center = 0; Xf = 300, Yf = 100; Counterclockwise = OFF

Arc1-Arc2 (C1,C2)	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 1	
X Centre	50
Y Centre	0
CCW	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 2	
X Centre	50
Y Centre	0
Relative	<input type="checkbox"/>
Xf	300
Yf	100
Zp	
Concordant Versus	<input type="checkbox"/>
<input type="checkbox"/> Speed	
Work Speed	

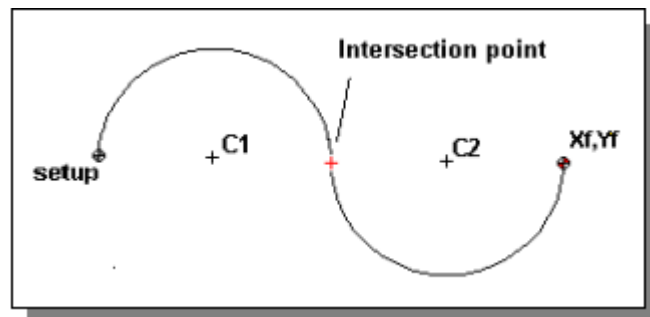


Fig. 3.19a : C1 Center is assumed relative to the Initial point,
C2 Center is assumed relative to the Final point Xf,Yf.

Available options:

- 1) Counterclockwise (for Arc1) : allows to select the Clockwise/Counterclockwise sense of the first arc
- 2) Concordant sense (for Arc2) : allows to select a Concordant or Opposite sense for the second

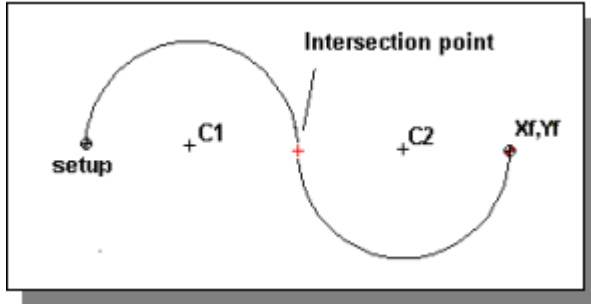


Fig. 3.19b : Options: Counterclockwise
/ Not Concordant

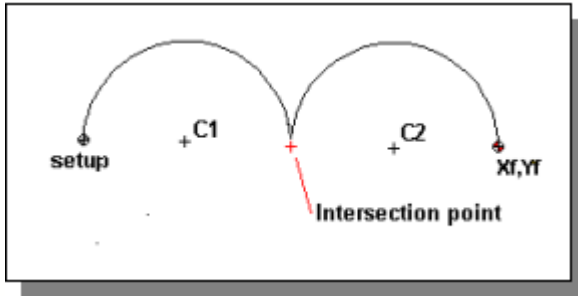


Fig. 3.19c : Options: Clockwise /
Concordant

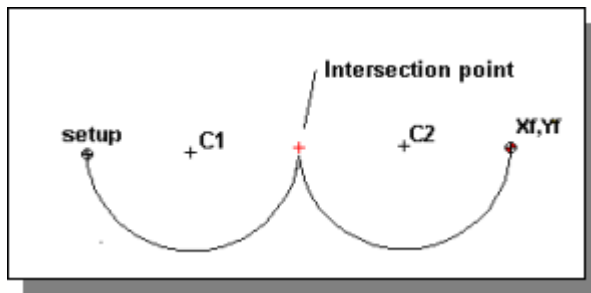


Fig. 3.19d : Options: Counterclockwise
/ Concordant

2.3.20 Double Arc "Arc1-Arc2 (C1,R2)"

Similar to the [Double Arc "Arc1-Arc2 \(C1,C2\)"](#): the second arc is defined by its Radius R2.

Then are required:

- the C1 Center of the first arc, X center, Y center;
- the final point of the second arc, Xf, Yf;
- the Radius **R2** of the second arc;

Assuming the Setup at the quotes (100,100,0), we select:

- **Arc1-Arc2 (C1,R2)**



entering the following parameters:

- Quote arc1 : X center = 50, X center = 0 , selection Counterclockwise = OFF
- Quote arc2 : Radius = 50, Xf = 300, Tf = 100, Zp = -10 , Option : Sense concordant: NOT

Arc1-Arc2 (C1,R2)	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 1	
X Centre	50
Y Centre	0
CCW	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 2	
Radius	50
Relative	<input type="checkbox"/>
Xf	300
Yf	100
Zp	-10
Concordant Versus	<input type="checkbox"/>
<input type="checkbox"/> Speed	
Work Speed	

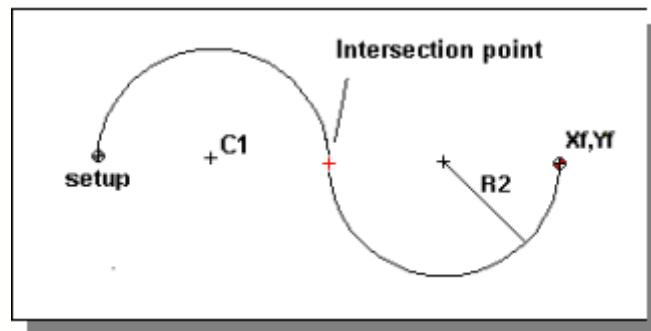


Fig. 3.20 : In the arc computing, the R2 radius substitutes the C2 center, that is derived automatically

2.3.21 Double Arc "Arc1-Arc2 (R1,C2)"

Opposite to the previous, in this function the double arc is defined by the Radius (R1) of the first arc and the Center (C2) of the second.

Then, the required parameters are:

- the Center of the second arc, X center, Y center;
- the Final point of the second arc, Xf, Yf;
- the Radius **R1** of the first arc;

Assuming a Setup position at the positions (100,100,0), we select

- Arc1-Arc2 (R1,C2)



entering the showed values for parameters (note Sense Not Concordant).

Arc1-Arc2 (R1,C2)	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 1	
Radius	50
CCW	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 2	
X Centre	-50
Y Centre	0
Relative	<input type="checkbox"/>
Xf	300
Yf	100
Zp	-10
Concordant Versus	<input type="checkbox"/>
<input type="checkbox"/> Speed	
Work Speed	

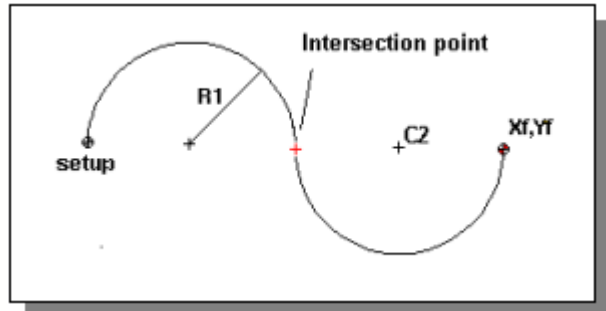


Fig. 3.21 : In this case the Center of the arc1 is computed on the base of the radius R1. The programming rules are similar to the previous cases.

2.3.22 Double Arc "Arc1-Arc2 (C1, exiting tangent)"

In this function the double arc is computed on the base of the Final point (Pf), the center (C1) of the first arc and the direction of the exiting tangent, for the second.

As usually, the exiting Tangent may be programmed directly, by the Angle, or implicitly, by the P3-P4 pair.

Assuming the Setup at the coordinates (100,100,-10), we select

- Arc1-Arc2 (C1, exiting tangent)



entering the following values for the parameters:

- Quotes arc 1 : X center = 50, Y center = 0 (Clockwise);
- Quotes arc 2 : Xf = 300, Yf = 100, Zp = -10 (in Absolute);
- Angle = 45°

Arc1-Arc2 (C1,tangent in exit)	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 1	
X Centre	50
Y Centre	0
CCW	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 2	
Relative	<input type="checkbox"/>
Xf	300
Yf	100
Zp	-10
Arc rot. direction	Automatic
<input type="checkbox"/> Final Angle	
X3	
Y3	
X4	
Y4	
Final Angle	45
<input type="checkbox"/> Speed	
Work Speed	

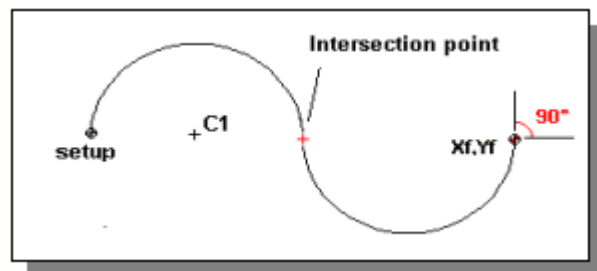
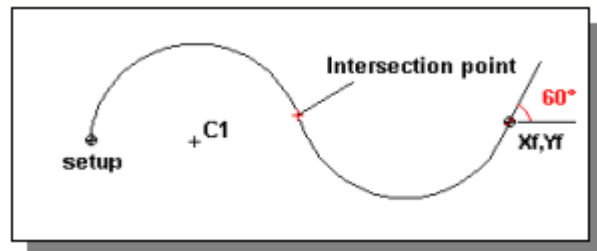
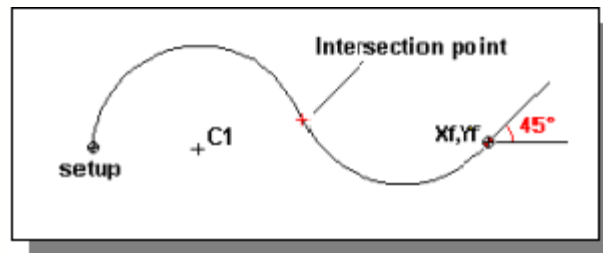


Fig. 3.22a

In this case, the exiting direction is programmed, directly, by the final Angle (= 45 degrees)

The other figures show the Double Arc modification, when the final Angle is programmed with different values (60 and 90 degrees).

→ Note: Programming by the P3-P4 points pair.

We can see, in alternative, the exiting direction programmed by the P3-P4 segment, that forces a 45° angle:

-X3 = 50, Y3 = 50;
-X4 = 100, Y4 = 100;

Arc1-Arc2 (C1,tangent in exit)	
Comment	
Exclusion	<input type="checkbox"/>
<div style="background-color: #f0f0f0; padding: 2px;"> Quotes arc 1 </div>	
X Centre	50
Y Centre	0
CCW	<input type="checkbox"/>
<div style="background-color: #f0f0f0; padding: 2px;"> Quotes arc 2 </div>	
Relative	<input type="checkbox"/>
Xf	300
Yf	100
Zp	-10
Arc rot. direction	Automatic
<div style="background-color: #f0f0f0; padding: 2px;"> FinalAngle </div>	
X3	50
Y3	50
X4	100
Y4	100
Final Angle	
<div style="background-color: #f0f0f0; padding: 2px;"> Speed </div>	
Work Speed	

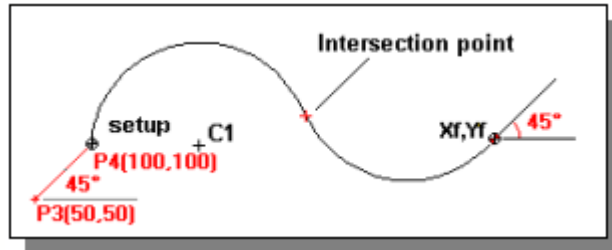
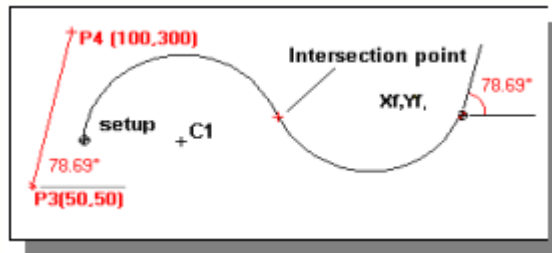


Fig. 3.22b:

In the following example, new values for the segment [P3=(50,50) - P4=(100,300)], will define a new angle of 78.69°.



2.3.23 Double Arc "Arc1-Arc2 (Entry tangent, C2)"

In this function, opposite to the [Arc1-Arc2 \(C1, exiting tangent\)](#), the Tangent is forced on the initial point, as usually by the Angle or by the P1-P2 points pair. The second arc is defined by the final point Pf and the Center C2.

Assuming the Setup at the coordinates (100,100,-10), we select

Arc1-Arc2 (entry tangent, C2)



and we enter the following parameters:

- X center = 0, Y center = -100 (coordinates are assumed in Relative respect to the Final point);
- Xf = 400, Yf = 300, Zp = -10
- Sense: Counterclockwise
- Angle = 70

Arc1-Arc2 (tangent in entry,C2)	
Comment	
Exclusion	<input type="checkbox"/>
☐ Quotes arc 1	
Arc rot. direction	Automatic
☐ Quotes arc 2	
X Centre	0
Y Centre	-100
Relative	<input type="checkbox"/>
Xf	400
Yf	300
Zp	-10
CCW	<input checked="" type="checkbox"/>
☐ Initial Angle	
X1	
Y1	
X2	
Y2	
Initial Angle	70
☐ Speed	
Work Speed	

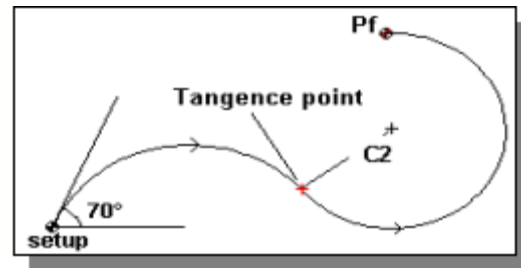


Fig. 3.23 : In the figure we see that the first Arc is programmed in CCW sense: with the Option "Automatic" system reverses the sense for the second, in order to maintain the Tangence continuity.

2.3.24 Double Arc "Arc1-Arc2 (Entry tangent, exit tangent)"

This function allows to program a Double Arc forcing the Entry (for the first) and the Exiting (for the second) Tangents.

Everyone of the two angles may be programmed or directly (Angle) or by the points pair: (P1-P2 for the first, P3-P4 for the second).

Assuming the Setup at the coordinates (100,100,-10), we select

- **Arc1-Arc2 (entry tangent, exiting tangent)**



entering the following values:

- **Arc 1** : Radius = 50 (sense = Clockwise);
- **Arc 2** : Xf = 400, Yf = 100 (sense = Counterclockwise);
- **Initial Angle** = 90;
- **Final Angle** = 45;

Arc1-Arc2 (tangent in entry,tangent in exit)	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 1	
Radius	50
CCW	<input type="checkbox"/>
<input type="checkbox"/> Quotes arc 2	
Relative	<input type="checkbox"/>
Xf	400
Yf	100
Zp	
Arc rot. direction	Automatic
<input type="checkbox"/> Initial Angle	
X1	
Y1	
X2	
Y2	
Initial Angle	90
<input type="checkbox"/> Final Angle	
X3	
Y3	
X4	
Y4	
Final Angle	45
<input type="checkbox"/> Speed	
Work Speed	

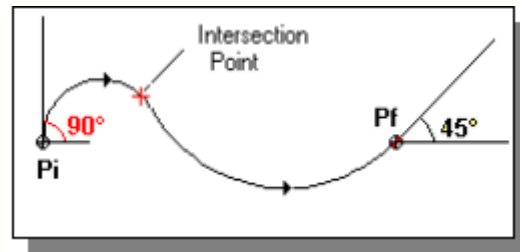


Fig. 3.24a :

The first arc is programmed with clockwise sense, the second with counterclockwise sense

2.3.25 Programming on sloped planes ("Oriented Mill Setup")

Making reference to the "[Oriented Mill Setup](#)" Macro-instructions, in this section the most important programming rules, concerning Workings using Rotating Tools, are described. In these operations we suppose that Tools are mounted on rotating heads, with the following movements capabilities: Rotation along the "Alfa" axis (primary rotating axis, oriented as Z) or Branding along the "Beta" axis (secondary rotating axis, assumed oriented as Y when Alfa=0) or together.

Workings with rotating tools may be programmed on the basic panel faces or on a sloped "fictitious" faces, programmed as described in Vol. 2 – Part 1 – Par. 2.1.4.

Since usually, in this second case, tool must be oriented "perpendicular" to the face itself (then following the "local Z" direction, as showed in the Fig. 3.25), system will propose automatically, when Setup is recalled, the corresponding values for:

- Alfa angle and
- Beta angle

as to assure perpendicularity to the face.

Programmer may modify these values, forcing a different tool direction, if required.

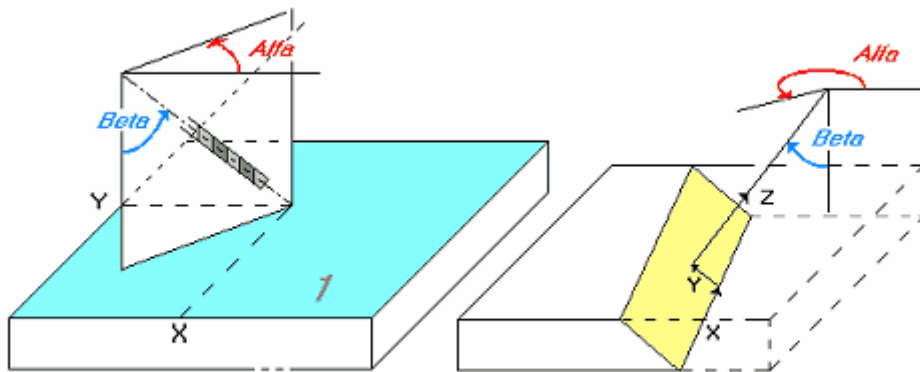


Fig. 3.25 - Oriented Mill Setup, on the Face 1 or on a sloped Face

Notice: Programmer must remember that, for the fictitious faces, system will assume, as programming Reference System, the following Cartesian group of axes:

- **Origin:** in the P0 point (the first programmed point, used for the face definition)
- **X Axis:** in the P0 – P1 direction
- **Z Axis:** Right Normal according to the counterclockwise sequence: P0 – P1 – P2
- **Y Axis:** Perpendicular to the previous two axes (see also Vol.2 – Part 1 – Par. 2.1.4)

Alfa angle: Convention positive if counterclockwise, with Alfa=0 when Tool has direction as X+

Beta angle: Local convention on the Branding plane: Beta=0 when the tool has a vertical (Z-) direction and Beta=90 when disposed horizontally.

Optional programming parameters for the "Oriented Mill Setup"

a) Others

→ Field: **Point Hooking**

Assumes the usual meaning, as previously described (see also Ex. 3.1 page 3.1).

→ Field: **Multiple Setup**

Assumes the usual meaning, as previously described (see also Ex. 3.1 page 3.2)

→ Field: **Interpolation 4° axis**

Available only in the Cartesian Reference, allows to program (on the face 1) Contouring workings where the tool is constantly rotated (along the "Alfa" axis) in order to maintain itself always "perpendicular" to the profile trajectory.

b) Feed

→ Field: **Entry Feed**

Assumes the usual meaning of the Tool "Entry speed", in the Z- direction.

→ Field: **Inserted Rounding feed**

Allows to force a slowing down, reducing feed in correspondence of the automatically inserted Roundings, where typically the trajectory is characterized by a lower curvature radius.

→ Field: **Spindle rev/min**

Assumes the usual meaning of Tool Rotating speed , in [rev/min].

c) Tools

→ Field: **Group**

Assumes the usual meaning of Group selection.

→ Field: **Tool 1**

Assumes the usual meaning of "First Tool" selection.

→ Field: **Tool 2**

Assumes the usual meaning of the eventual second Tool selection.

Assume l'usuale significato di selezione dell'eventuale secondo Utensile di lavorazione.

d) Mill Radius

→ Field: **Mill radius value**

Allows to enter, locally, the Mill Radius Correction value, if different from the selected tool radius.

→ Field: **Mill Radius Correction**

Allows to select between the modes: None – Left – Right.

[] **Progressive Mill Radius Correction start-up**

With this option, Programmer may select a progressive mill radius correction: in this case, the Setup point is not corrected, since correction will be applied only on the following point (then the final point of the first geometric element): then the trajectory will cause a progressive correction application.

[] **Progressive Mill Radius Correction closing**

Similar to the previous, is referred to the end of profile: the mill radius correction is removed from the last profile point, also in this case causing a particular trajectory between penultimate and final point.

e) M Fields

These Options allows to enter a required values for the M1, M2 and M3 variables, used in the PLC procedures.

2.4 Macro

2.4.1 "Circle (center)" macro

This Macro allows to program a complete circle, starting from the Current point, only with the Center and rotation Sense defining.

Circle (center)	
Comment	
Exclusion	<input type="checkbox"/>
📁 Quotes	
Zp	-10
Relative	<input type="checkbox"/>
📁 Other Quotes	
CCW	<input type="checkbox"/>
Y Centre	100
X Centre	100
📁 Speed	

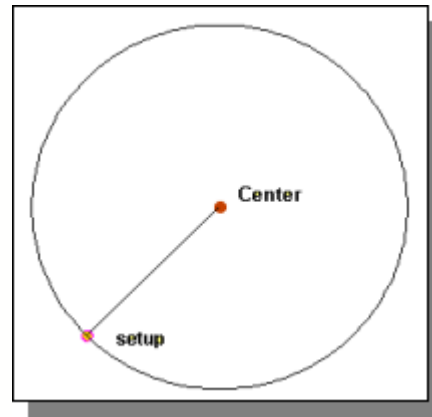


Fig. 4.1

Center, in this function, is always assumed in Relative mode respect to the Initial point

Assuming the Mill Setup in the same position of the Ex. 3.0, we select:

Circle(center)



The following parameters are required:

- Center coordinates : X center and Y center, (assumed in Relative mode).
- Depth: Zp
- Sense: CW or CCW

Execution performs a complete circle, returning to the initial point.

2.4.2 "Circle (three points)" macro

Similar to the "Circle (center)", with this function the circumference is defined by 3 points (not aligned).

Using the same Setup, we select:

- **Circle(three points)**



In the Parameters Table, the two other points P1 (X1,Y1) and P2 (X2,Y2) are required: the first point is assumed on the current.

In the example, the following values are entered :

- X1 = 50, Y1 = 100,
- X2 = 100, Y2 = 50 (in Relative)
- Zp = -10

Circle (three points)	
Comment	
Exclusion	<input type="checkbox"/>
<input checked="" type="checkbox"/> Quotes	
Relative	<input checked="" type="checkbox"/>
Zp	-10
<input checked="" type="checkbox"/> Other Quotes	
CCW	<input type="checkbox"/>
Y2	50
X2	100
Y1	100
X1	50
<input checked="" type="checkbox"/> Speed	

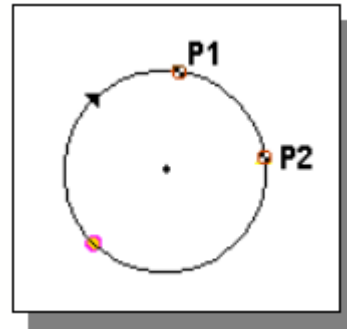


Fig. 4.2

Initial point: (100,100)

P1 = (150,200)

P2 = (200,150)

Note: In this case, the moving sense (CW or CCW) is automatically deduced by the system, according to P1 and P2 sequence.

2.4.3 "Oval" Macro

This function allows to build a special profile, realized as an approximate Ellipsis: approximation is performed with 4 different arcs, symmetrically disposed respect to the Center, with the same radius for the opposite arcs. The minimum radius Arc (near to the Focus) is directly programmed whereas the greater arc is automatically computed, on the base of the programmed axes.

Setup is automatically assumed in the lower position, as showed in the Fig. 4.3

Selecting the Macro :



the following parameters are required:

- Center position: X center = 300, Y center = 200,
- Depth: Zp = -10,
- Semi-axes length : X-axis = 300, Y-axis = 120,
- Radius of the minimum arc : R = 59.

Oval	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
X Centre	300
Y Centre	200
Axis X	300
Axis Y	120
Zp	-10
Relative	<input type="checkbox"/>
Radius	59
Oval Portion	Entire
CCW	<input type="checkbox"/>
<input type="checkbox"/> Other Quotes	
Entry Arc	<input type="checkbox"/>
Exit Arc	<input type="checkbox"/>
External entry/exit arc	<input type="checkbox"/>
Entry radius	
Mirror X	<input type="checkbox"/>
Mirror Y	<input type="checkbox"/>
Angle	
Point link	<input type="checkbox"/>

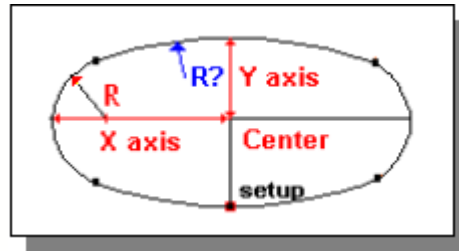


Fig. 4.3a

In the figure, the **R** radius defines the smaller arc. The larger arc radius **R?** is automatically computed by the system.
The Setup position is assumed by default.

→ Option: **Relative**

With the usual meaning, is referred only to the Center programming.

→ Option: **Counterclockwise**

Defines the rotating sense for working (may be affected by following mirroring operations).

→ Option: **Oval portion**

Allows to select if execution must include the complete oval profile, or $\frac{1}{2}$ (Half oval, from Setup to the diametrically opposite point) or $\frac{1}{4}$ (only a quadrant).

→ Option: **Entry arc**

allows to move the Setup, adding an entry semi-circular path (tangent to the original initial point), where the Radius may be programmed [Entry Radius], as showed in the fig. **4.3b**.

→ Option: **Exiting Arc**

Similar to the previous (with the same Radius), but for exiting.

In the example of the fig. 4.3b, Oval is programmed complete, with an Entry arc with radius = 20 mm.

Oval	
Comment	
Exclusion	<input type="checkbox"/>
<input checked="" type="checkbox"/> Quotes	
X Centre	300
Y Centre	200
Axis X	300
Axis Y	120
Zp	-10
Relative	<input type="checkbox"/>
Radius	50
Oval Portion	Entire
CCW	<input type="checkbox"/>
<input checked="" type="checkbox"/> Other Quotes	
Entry Arc	<input checked="" type="checkbox"/>
Exit Arc	<input type="checkbox"/>
External entry/exit arc	<input type="checkbox"/>
Entry radius	20
Mirror X	<input type="checkbox"/>
Mirror Y	<input type="checkbox"/>
Angle	
Point link	<input type="checkbox"/>

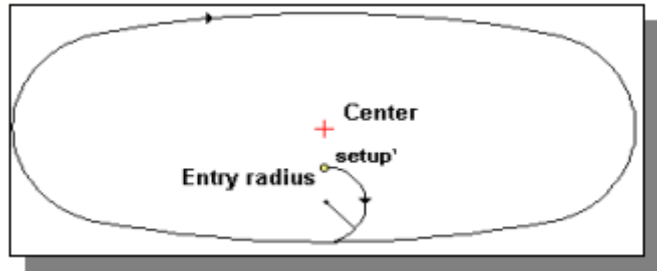


Fig. 4.3b: With the Entry Arc option, program moves the original Setup point in the new **Setup'** position, according to the Entry radius.

→ Option: **Mirror X**

Reverses the path with an horizontal symmetry, then also reversing the original moving sense, as showed in the fig. 4.3c.

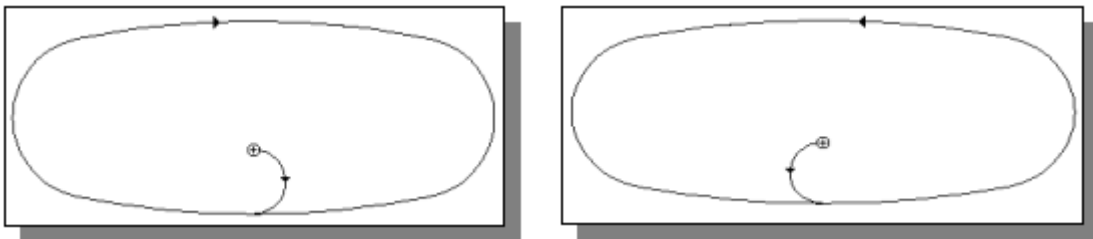


Fig. 4.3c: Mirror X option.

→ Option. **Mirror Y**

Vertical mirroring, as showed in fig. 4.3d. Also in this case, the vertical symmetry causes a sense reversing.

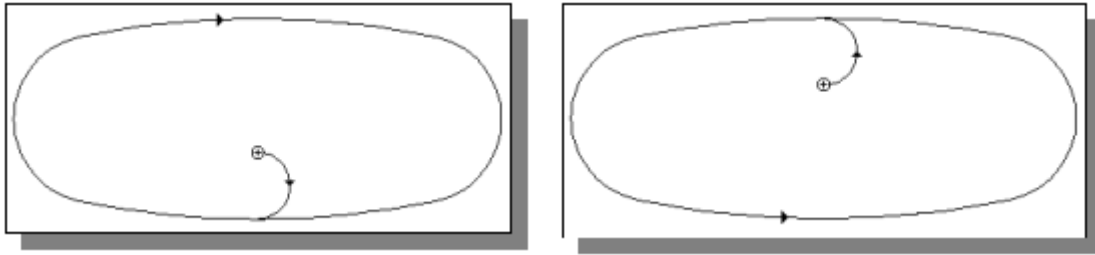


Fig. 4.3d: Mirror Y option.

2.4.4 "Ellipsis X" (Ellipsis Y) macro

This function allows to build a more precise Ellipsis since this geometrical function is approximate by a large number of arcs (programmable). Ellipsis is completely defined by its Center and by the two Semi-axes:

- Axis 1 in X direction and
- Axis 2, in Y direction.

Note : As in the previous case, the Ellipsis may be rotated in the XY plane, with a programmable Angle parameter: in effect, the Ellipse Y (where the larger semi-axis is disposed in Y direction) may be considered as an Ellipsis X with a rotating Angle = 90°

Selecting the macro :

- **Ellipsis X**



the following parameters must be entered:

- 1) Center position: X center = l/2, Y center = h/2,
- 2) Depth : Zp = -10,
- 3) Dimensions: Axis 1 = 200, Axis 2 = 80,
- 4) Approximating : Number of arcs = 10 (warning: this value is referred to a single quadrant, equivalent to $\frac{1}{4}$ of the complete ellipsis!).
- 5) Rotation : Angle = ...

Ellipse x	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
X Centre	l/2
Y Centre	h/2
Axis X	200
Axis Y	80
Zp	-10
Relative	<input type="checkbox"/>
Ellipse Portion	Entire
CCW	<input type="checkbox"/>
<input type="checkbox"/> Other Quotes	
Entry Arc	<input type="checkbox"/>
Exit Arc	<input type="checkbox"/>
External entry/exit arc	<input type="checkbox"/>
Entry radius	
Mirror X	<input type="checkbox"/>
Mirror Y	<input type="checkbox"/>
Angle	
Point link	<input type="checkbox"/>
Number of arcs	8

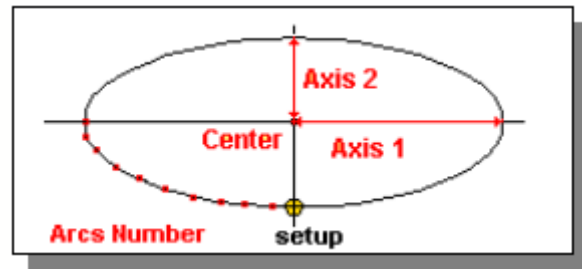


Fig. 4.4: Precision of the ellipsis rebuilding is improved increasing the Number of the arcs.

→ **Options:**

The available Options have the same meaning as in the "Oval" macro.

→ **Macro: "Ellipsis Y"**

Similar to the present, simply assumes the larger Axis in Y direction (then is equivalent to an Ellipsis X rotated of 90 degrees).

2.4.5 "Polygon inscribed in a Circle" macro

This macro allows to build a milling profile including a regular polygon, inscribed in the programmed circumference. The sides number is also programmable.

Setup is automatically assumed on the first vertex of the polygon (defined by the initial Angle).

Selecting:

-Polygon inscribed in a Circle



the following parameters must be entered:

- 1) Circumference Center : X center = l/2, Y center = h/2,
- 2) Depth : Zp = -10,
- 3) Radius of the circumference : Radius = 150
- 4) Number of sides : Sides = 6 ,
- 5) First vertex Angle (=Setup) : Angle = 0,

Polygon inscribed in a circle	
Comment	
Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Quotes	
X Centre	l/2
Y Centre	h/2
Zp	-10
Relative	<input type="checkbox"/>
Radius	150
Sides	6
Angle	0
Point link	<input type="checkbox"/>
CCW	<input type="checkbox"/>
<input type="checkbox"/> Tools	
Tool	
Tool radius compensation	None
Tool radius value	

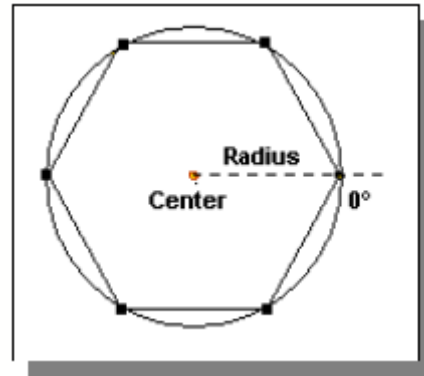


Fig. 4.5: Il valore zero dell'inclinazione è riferito al punto di Setup; modificando tale angolo il poligono ruoterà in senso antiorario. Tutti i vertici del poligono ricadono sulla circonferenza programmata

- Option: **Relative**

Is referred only to the circumference Center programming.

- Option: **Counterclockwise**

Is referred to the moving sense, during execution.

- Option: **Point hooking**

Is referred, as usually, to the Setup: polygon may be hooked to the previous profile.

2.4.6 "Polygon circumscribed to a Circle" macro

The Polygon is circumscribed to the programmed circumference, then its sides are tangent on the contact point

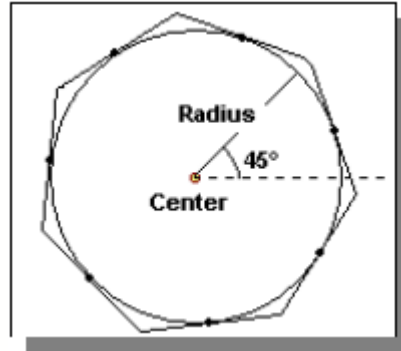
Selecting:

- **Polygon circumscribed to a Circle**



we find the same parameters and the programming rules and Options of the case "[Polygon inscribed in a Circle](#)".

Polygon circumscribed to a circle	
Comment	
Exclusion	<input type="checkbox"/>
☞ Quotes	
X Centre	l/2
Y Centre	h/2
Zp	-10
Relative	<input type="checkbox"/>
Radius	150
Sides	6
Angle	0
Point link	<input type="checkbox"/>
CCW	<input type="checkbox"/>



2.4.7 "Fillet" macro

This function allows to introduce a fillet on a programmed corner, defining only the Radius. System computes automatically the intersection points as to maintain continuity of Tangent.

The macro is selected by the button:



Programming requires to define the "theoretical" corner position and characteristic and the entry and exiting sides: the smoothing round is characterized by the Radius.

The entry segment is defined by the current (initial) point and the Corner, the second segment by the corner and the programmed P4 point.

→ Note: In alternative to the P4 point, the exit segment direction may be directly programmed by the "Final Angle" value.

Then, the parameters to be programmed are:

- 1) Corner position : X corner = .. Y corner = ..
- 2) Point P4: X4 = .., Y4 = .. (or Final Angle = ..)
- 3) Radius of rounding : Radius = ..

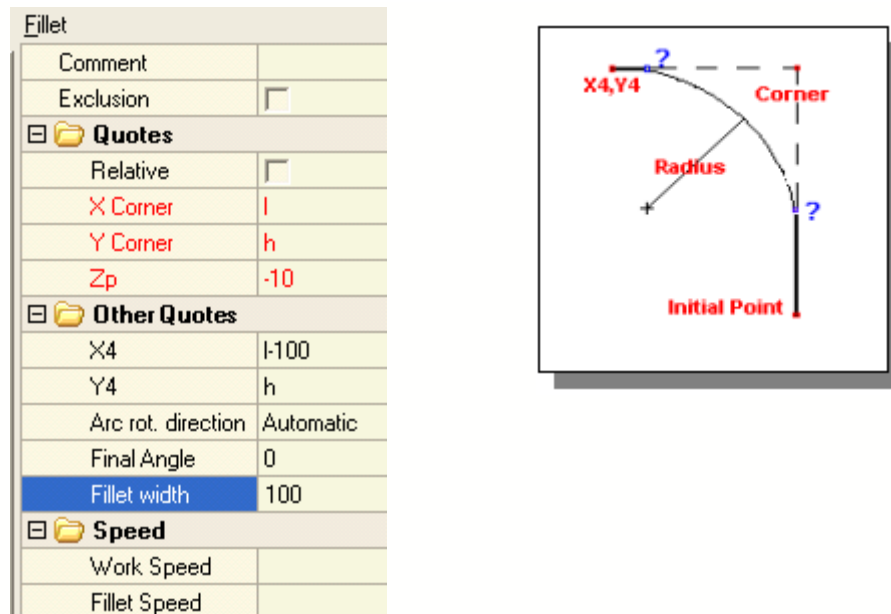


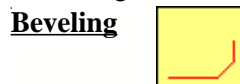
Fig. 4.7: The arc intersection points (indicated as ?) are automatically computed, to maintain continuity of tangence

2.4.8 "Beveling" macro

This function allows to introduce a linear Beveling on a programmed corner.

Corner is programmed with the same rules as the "[Corner Rounding](#)" function.

Beveling is recalled with the button:



Beveling is a segment that cuts the programmed corner, connecting two points, internal to the entry and exiting segment, at a programmed distance from the corner itself.

If we enter the following values in the Parameters Table:

1. X corner = l, Y corner = h,
2. X4 = l-100, Y4 = h,
3. Beveling length = 100

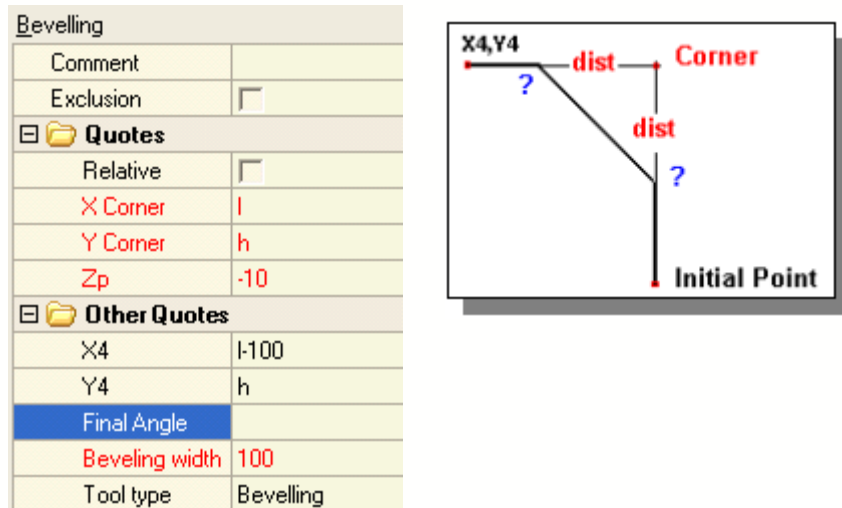


Fig. 4.8: The intersection points (marked by ?) are automatically computed by the system.

Also in this case, the exiting segment may be defined by the "Final Angle".

2.5 Subroutines

2.5.1 Subroutine "sub1.cnc" programming

In this paragraph we can see the Subroutine mechanisms for creating and storing, before to analyze the different recalling methods.

In the following, the different recalling functions and the Subroutine application rules will be described.

The present example will describe the Subroutine creation: when completed, any Subroutine must be saved in a dedicated Sub-Directory, normally named **Product\Sub**. In this case, the subroutine is very simple, since includes a single Drill, programmed on the Face 1, and will be saved with name **sub1.cnc**.

This subroutine will be programmed in "Parametric" mode, using the following Variables:

- Drill X quote = parameter **r0**
- Drill Y quote = parameter **r1**
- Depth = -10 (fixed)
- Diameter = 20 (fixed)

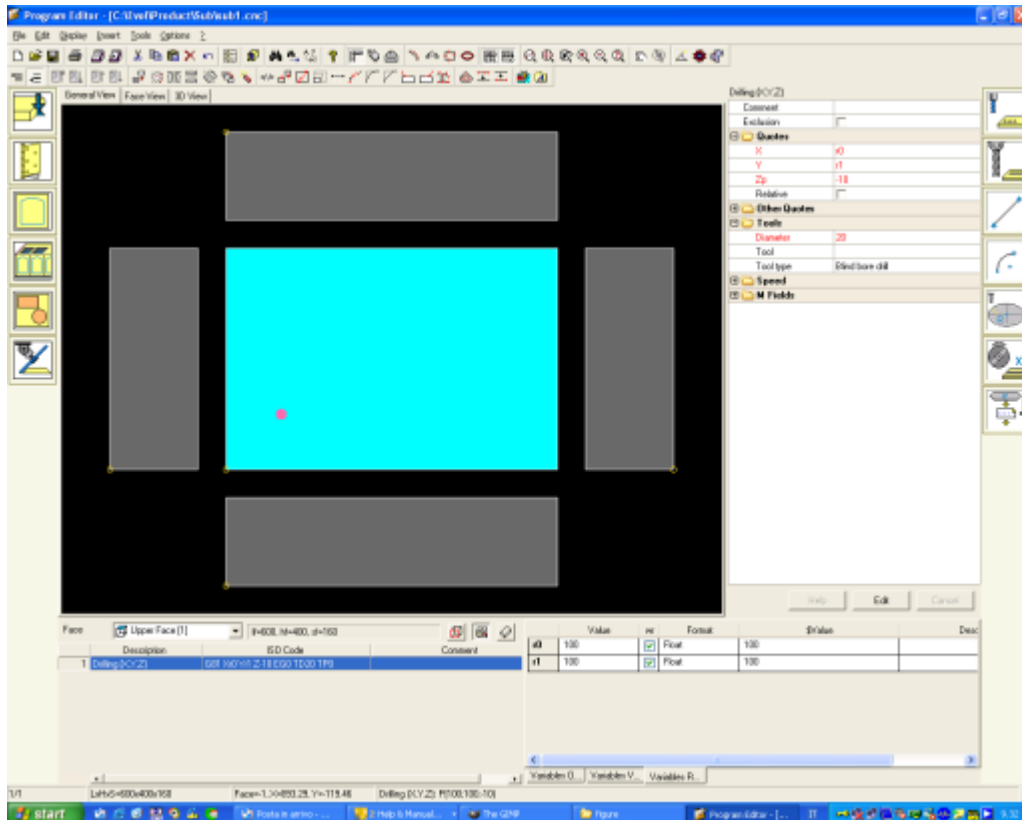


Fig. 5.1a : Subroutine "sub1" : Programming Table for DRILL (X,Y,Z)

The default values for the internal Variables may be assigned in a dedicated Data Table (showed in the Fig. 5.1b), may be accessed clicking the corresponding button of the Tool Bar (see also Button (1) indicated in the [Fig. 5.6](#) - Appendix 2).

In this case, the same value (then $r0=100$, $r1=100$) assigned to the two Variables.

Since these internal variables are supposed to be modified by the recalling main program, it must be marked as "re-assignable", selecting the [x] status in the "**wr**" column.

The variables $r0$ and $r1$, programmed in this example, are used to code the application quotes of the drill: consequently may support signed and fractional values, then in the "**Format**" field, the "FLOAT" typology must be selected.

In the **\$Value** field the assigned value is yet presented, in "string" format while in the "Description" field Programmer may enter a general purpose string ("comment") to indicate, for instance, the logical meaning of the used variables.

	Value	wr	Format	\$Value
r0	100	<input checked="" type="checkbox"/>	Float	100
r1	100	<input checked="" type="checkbox"/>	Float	100
r2	0	<input type="checkbox"/>	Float	
r3	0	<input type="checkbox"/>	Float	
r4	0	<input type="checkbox"/>	Float	
r5	0	<input type="checkbox"/>	Float	
r6	0	<input type="checkbox"/>	Float	
r7	0	<input type="checkbox"/>	Float	
r8	0	<input type="checkbox"/>	Float	
r9	0	<input type="checkbox"/>	Float	

Fig. 5.1b - Variables Data Table for the Subroutine "sub1.cnc"

2.5.2 Subroutine "sub2.cnc" programming

This new example describes a new other subroutine programming: this new (will name "sub2.cnc") includes a single Drill, programmed on the face 4, where also the Diameter has been entered in parametric mode, then:

- X quote = parameter **r0**
- Y quote = parameter **r1**
- Diameter = parameter **r2**

The following figures show the programming environment for the subroutine "sub2" (Fig. 5.2a) and the corresponding Variables Data Table assigning (Fig. 5.2b).

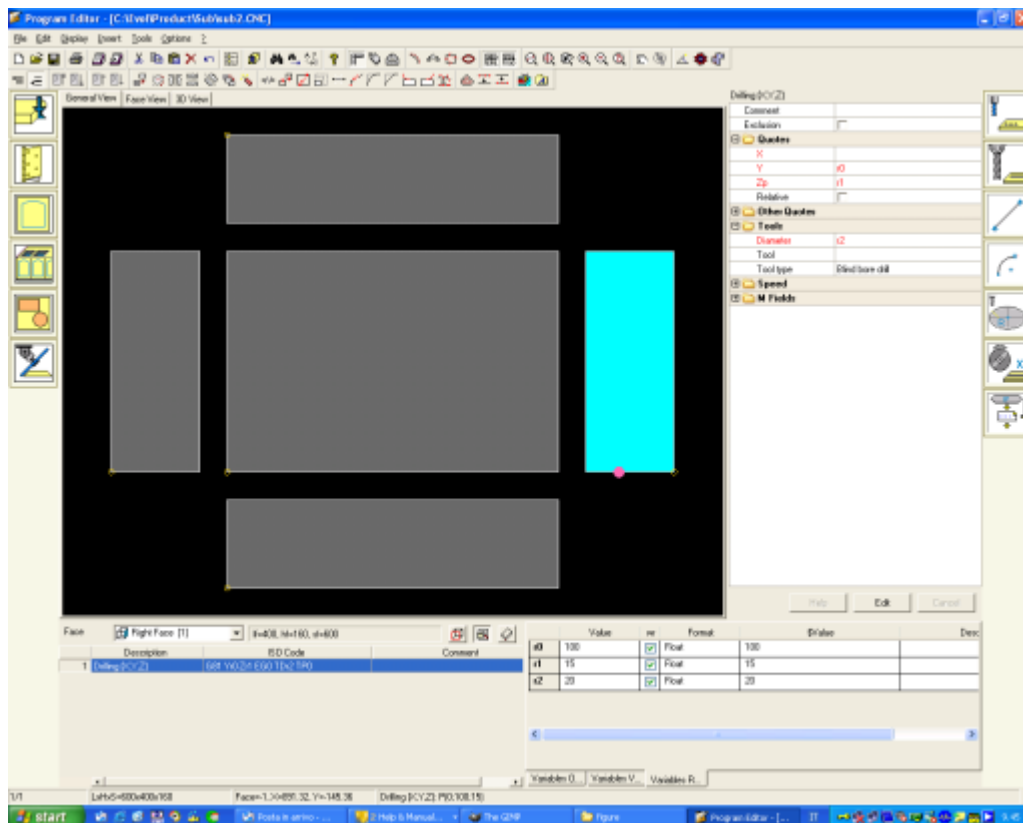


Fig. 5.2a - Subroutine "sub2.cnc".

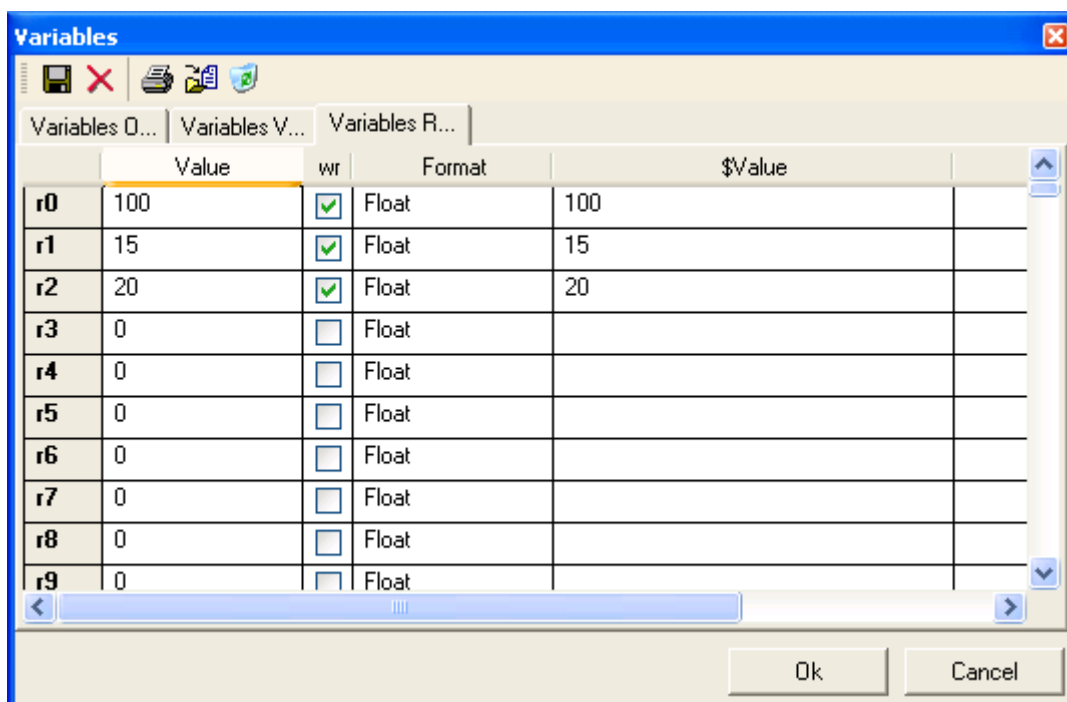


Fig. 5.2b - Variables Data Table assigning for "sub2.cnc".

2.5.3 Subroutine recall with the "SUB0" instruction

We can see now, in this example, the subroutines recalling methods, during main program creating.

The recalling mechanism is based on two different functions, the "SUB0" and "SUB2" instructions that may be selected by the corresponding buttons of the Workings Palette (indicated in the area (3) of the Fig. 5.6 - Appendix 2).

We remember that these two functions are similar, only changing the "Multiplying" mechanism:

- **SUB0** : multiplying mechanism occurs along a sloped line, and the inter-axis between the application points is defined by the "Offset X" and "Offset Y" parameters (more, a progressive change of depth may be introduced by the "Offset Z" value)
- **SUB2** : multiplying is realized on a reticular pattern, defined by the "rows" and "columns" numbers and by the corresponding inter-axis (yet named Offset X and Offset Y)

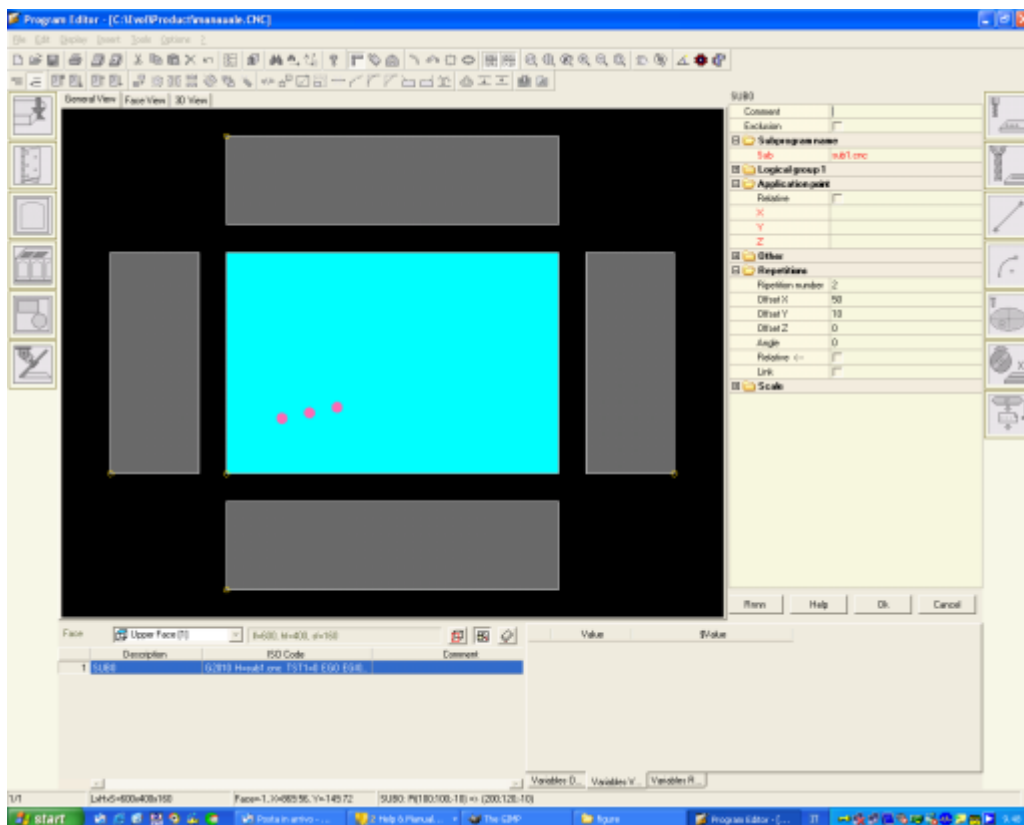


Fig 5.3a- ISubroutine "sub1.cnc" multiple recalling by the SUB0 function

In this example the SUB0 function is used for "sub1.cnc" recalling.

With reference to the Fig. 5.3, we can see the most important rules for the different fields programming and the options selecting.

→ Field : **Subroutine Name**

Allows to select the required subroutine, by direct entering or by List box selection

→ Option: **Logical Group 1**

Not used in this example, allows to define a Logical condition to enable (if TRUE) or not (if FALSE) the subroutine recall.

The logical condition is defined comparing the "Expression1" with the "Expression2" (including numbers or mathematical formulas based on the r0-r299 Variables of the Main program) with the use of one of the following "compare" Operators (<, >, =, <=, =>, <>)

→ Field: **Application point**

Indicates (if cannot be implicitly deduced by the "parametric" quotes expression) the application quote of the subroutine, assumed as the point where the Subroutine reference point (the first programmed point) must be translated.

As usually, the X, Y and Z quotes may be programmed in Absolute or in Relative [] mode.

→ Option: **Angle**

Allows to rotate the subroutine layout when recalled, assuming the application point as fulcrum.

→ Option: **Face**

We must consider that, normally, a Subroutines may include many workings, programmed on different faces. Then, when it is recalled from a Main program, there are two possibilities:

- first, the main program requires to import all the workings (of all the faces), applying them on the same faces: in this case the "Face" code must not be programmed
- alternatively, the main program may import the workings only for a then the If missing, the subroutine is completely imported and all the programmed workings, included in the subroutines (for every face), are inserted in the corresponding faces of the main. Alternatively, if a specific face is programmed, only the subroutines instructions, programmed on this face, will be imported.

→ Option: **Relative <--**

If the Relative [x] option is selected, relatively to the application point, the corresponding quotes will be considered as incremental and referred to the application point (then used as Origin) of the previous bloc (immediately before recalling).

→ Option: **Hooking**

very convenient in the case of milling profiles, this option allows to "link" the recalled path to the final point (current point) of the previous profile. Obviously, the original Setup, in the subroutine, is erased to maintain the path continuity.

→ Option: **Reversing**

Also this option is referred to a milling profile: in the application in the main program, the subroutine path is reversed, exchanging the Setup point with the original Final point.

→ Option: **Mirror X / Mirror Y**

Allows the subroutine layout mirroring, introducing respectively an horizontal (in X) or a vertical (in Y) symmetry operation.

Repeating Fields

→ Field: **Repeat Number**

Allows to introduce a Multiplying factor, in order to repeat the subroutine application, over the basic, for a programmable number of times. The Repeating logic is different for the two recalling instructions: "SUB0" or "SUB2".

In the example, (where SUB0 has been used) two additional repeating are programmed, then causing a total of three different subroutine applications.

→ Options: **Offset X / Offset Y / Offset Z**

In the case of SUB0, these values define (the first two) the components along X and Y of the multiplying inter-axis vector (in the example, component X = 50 mm, the component Y = 10 mm.).

The Offset Z parameter allows to increase or reduce progressively the application depth.

→ Option: **Relative** <--

If this option is selected, the current Offset are considered starting from the application point of the previous repetition, then considered as origin.

→ Option: **Hooking**

Allows to link the current Repetition to the final point of the previous.

Scale Factor

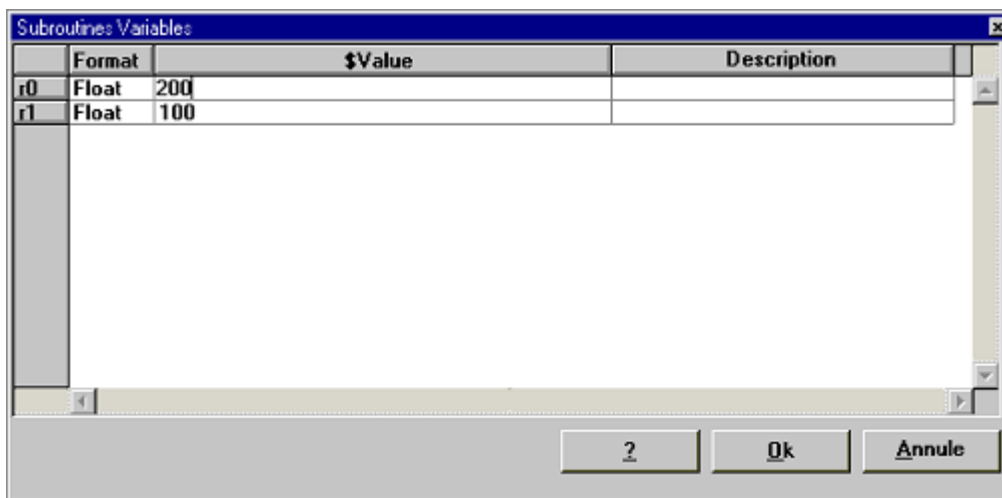
→ Option: **Scale enable**

Enables the Scale Factor application.

→ Option: **Scale Multiplier**

Defines the value of the Increasing (if > 1) or Reducing (if < 1) factor to be applied to the Subroutine layout.

In the present example, as showed in the Fig. 5.3b, the original **r0** parameter has been re-assigned with the new value (=200): the Data Entry table, for re-assigning, may be open with the "**Rnnn**" button.



	Format	\$Value	Description
r0	Float	200	
r1	Float	100	

Fig. 5.3b - Re-assigning Data Table, for the "sub1.cnc" variables

2.5.4 Subroutine recalling with the "SUB2" function

In this case we suppose (starting from the Main program as developed in the previous example), to introduce a new recalling, in this case of the subroutine "sub2.cnc" (as defined in the [Ex. 5.2](#)), using the other function, the **SUB2**.

This last, as described, differs from the previous only for a different Multiplying mechanism, that, in this case, includes the possibility to define a reticular pattern, as showed in the Fig. 5.4a.

Note: As may be noted, in this case the "Face" option has been used to select, for importing, only the workings programmed, in the subroutine, on the [face 4](#) (that is, in particular, the only programmed face) but requiring to insert them on the Face 1 of the Main.

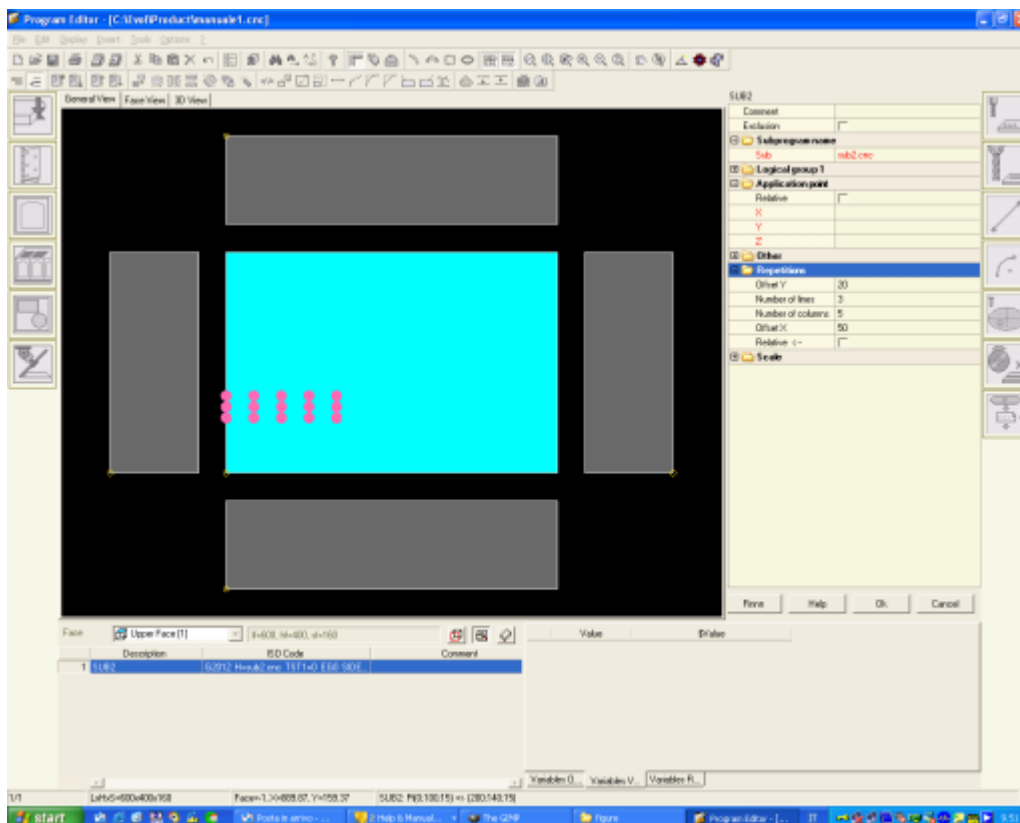


Fig. 5.4a - Importing with the SUB2 function

In particular, respect to the previous case, we can see the different programming rules concerning the Repeating modalities:

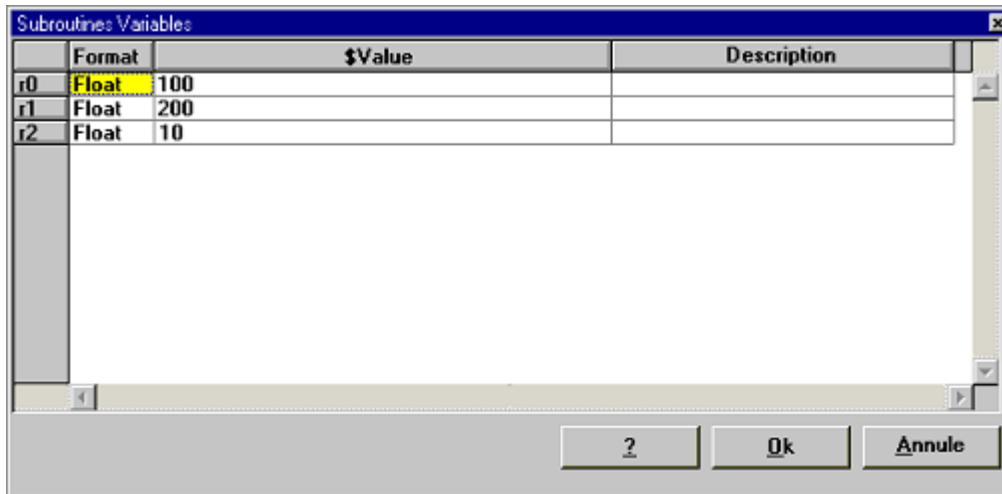
→ Option: **Rows Number / Columns Number**

Defines the Reticule structure, in terms of number of rows (= 3) and columns (= 5).

→ Option: **Offset X / Offset Y**

In this case these parameters define, respectively, the X inter-axis (distance between columns, = 50 mm.) and the Y inter-axis (distance between rows, = 20 mm.).

Also in this case, we have supposed that the recalling program force a new value to the subroutine variables, re-assigning the Y application quote (r1, forced to 200 mm.) and the Diameter (r2, forced to 10 mm.), as showed in the Fig. 5.4b.



	Format	\$Value	Description
r0	Float	100	
r1	Float	200	
r2	Float	10	

Fig. 5.4b - Data Table for "sub2.cnc" variables reassigning

As result we will have: 15 holes (5 rows x 3 columns), with diameter = 10 mm (r2), with inter-axes of 50 mm (in X) and 20 mm. (in Y).

The first drill will be positioned to the programmed "application" point, then 200,100 (r0,r1).

2.5.5 Appendix 1: Data Entry tables for the functions: SUB0 and SUB2

The complete Data Entry tables, for the recalling functions - SUB0 and SUB2 - are here resumed:

SUB0		SUB2	
Comment		Comment	
Exclusion	<input type="checkbox"/>	Exclusion	<input type="checkbox"/>
<input type="checkbox"/> Subprogram name		<input type="checkbox"/> Subprogram name	
Sub	sub1.cnc	Sub	sub1.cnc
<input checked="" type="checkbox"/> Logical group 1		<input checked="" type="checkbox"/> Logical group 1	
<input type="checkbox"/> Application point		<input type="checkbox"/> Application point	
Relative	<input type="checkbox"/>	Relative	<input type="checkbox"/>
X		X	
Y		Y	
Z		Z	
<input type="checkbox"/> Other		<input type="checkbox"/> Other	
Angle		Angle	
Face		Face	
Relative <--	<input type="checkbox"/>	Relative <--	<input type="checkbox"/>
Link	<input type="checkbox"/>	Link	<input type="checkbox"/>
Inversion	<input type="checkbox"/>	Mirror X	<input type="checkbox"/>
Mirror X	<input type="checkbox"/>	Mirror Y	<input type="checkbox"/>
Mirror Y	<input type="checkbox"/>	<input type="checkbox"/> Repetitions	
<input checked="" type="checkbox"/> Repetitions		Offset Y	
Ripetition number	0	Number of lines	
Offset X	0	Number of columns	
Offset Y	0	Offset X	
Offset Z	0	Relative <--	<input type="checkbox"/>
Angle	0	<input type="checkbox"/> Scale	
Relative <--	<input type="checkbox"/>	Scale multiplier	0
Link	<input type="checkbox"/>	Enable Scale	<input type="checkbox"/>
<input type="checkbox"/> Scale			
Enable Scale	<input type="checkbox"/>		
Scale multiplier	0		

Fig. 5.5 - Data Entry tables for SUB0 and SUB2.

- **Description**: "Comment" field, for the subroutine description (free characters string)
- **Subroutine Name**: Name of the recalled subroutine
- **Logical Group 1**: Logical condition to enable (if TRUE) or not (if FALSE) recalling.
- **Application Point**: X,Y Application quotes of the subroutine (in Absolute or Relative mode): at this point will be referred the first programmed point (into the subroutine), that will be considered as reference.
- **Angle**: Rotation Angle (optional) for the subroutine application (the programmed rotation will assume, as center, the "Application" point.
- **Face**: Allows to import the entire subroutine (all programmed faces will be imported and inserted in the same of the main), if "no value" is entered or, alternatively, if a value is specified, only the corresponding face of the subroutine will be imported, and inserted in the current main face.
- [] **Relative <--**: If programmed (Relative [x]) on the application point, this new will be considered in incremental mode, using as reference, the initial application point of the previous

Working.

- [] Link: Allows to "link" the subroutine path to the final point of the previous profile.
- [] Reversion: Allows to reverse the path direction of the subroutine profile, exchanging its original Setup point with the final.
- [] Mirror X / Mirror Y: Allows to mirror the subroutine layout, introducing, respectively, an horizontal (in X) or vertical (in Y) symmetry.

Repeat for SUB0

- Repeat number: Indicates how many additive application of the subroutine, over the base, must be performed
- Offset X / Offset Y / Offset Z: components of the "inter-axis" vector
- Angle: Defines the incremental angle to be use for the repeated applications of the subroutine
- [] Relative<--: if enabled, the Offsets will be referred to the application point of the previous Repeat.
- [] Link: If enabled, the new pattern is hooked to the final point of the previous Repeat.

Repeat for SUB2

- Rows Number / Columns Number: Indicate the reticular pattern structure for repeating
- Offset X / Offset Y: Indicate the distance between columns and rows.
- [] Relative<--: if enabled, the Offsets will be referred to the application point of the previous Repeat.

Scale Factor

- [] Scale enable: Scale factor enable
- Scale Multiplier: Defines the increasing (if >1) or reducing (if <1) factor, for the subroutine layout.

2.5.6 Appendix 2: Toolbar and Palette

In the Fig. 5.6 are showed the most important button related to the subroutines management:

Area 1 = Button for the **r..** variables recalling, for the original assigning during the subroutine creation.

Area 2 = **Rnnn** button, for the r0-r299 variable re-assigning, at the moment of recalling.

Area 3 = Buttons for the **SUB0** and **SUB2** functions selection

Area 4 = Button for the **r0-r299** variables recalling, for displaying.

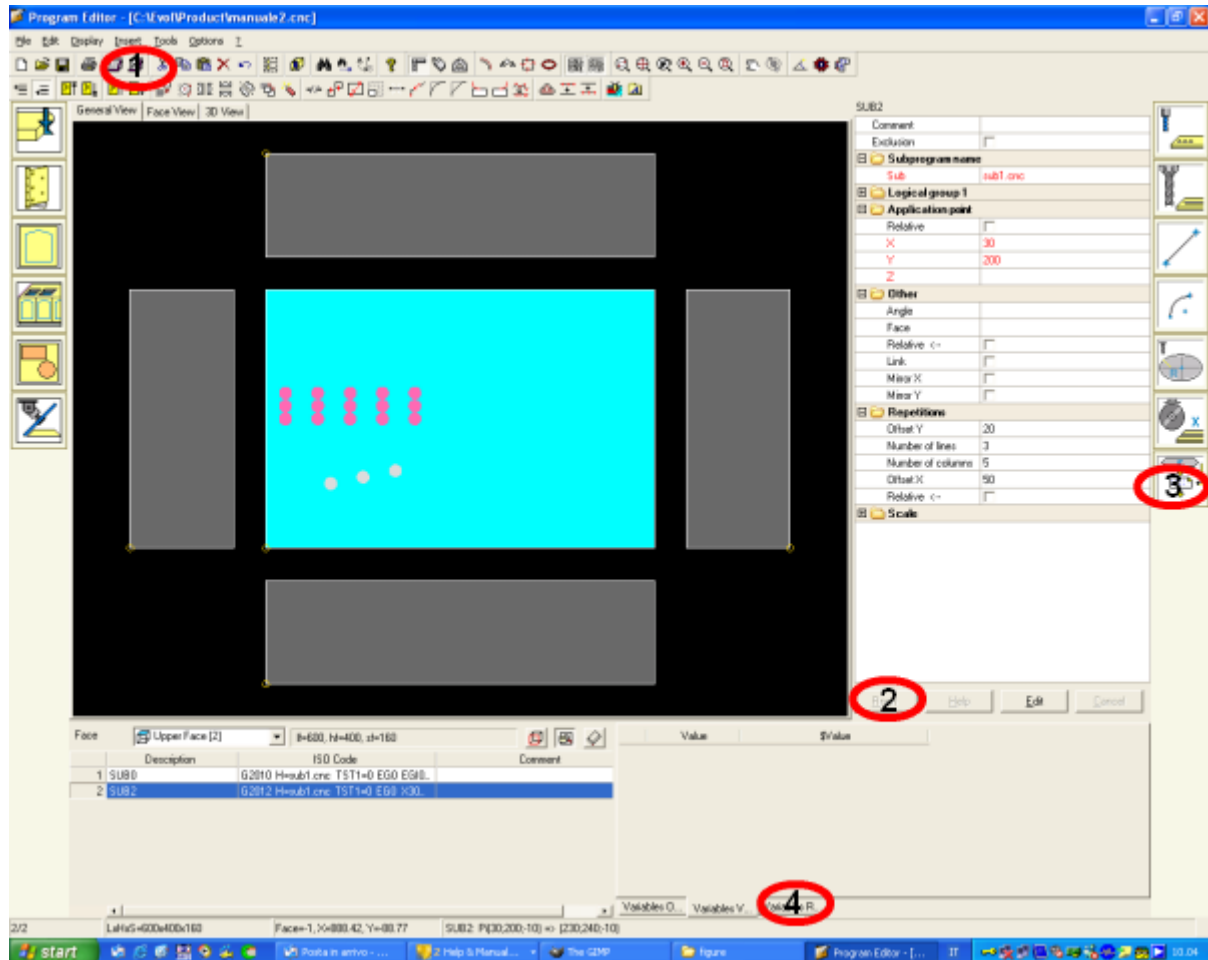



Fig. 5.6 - Subroutine management buttons

2.6 Surface Clearing and Develop Text Tools

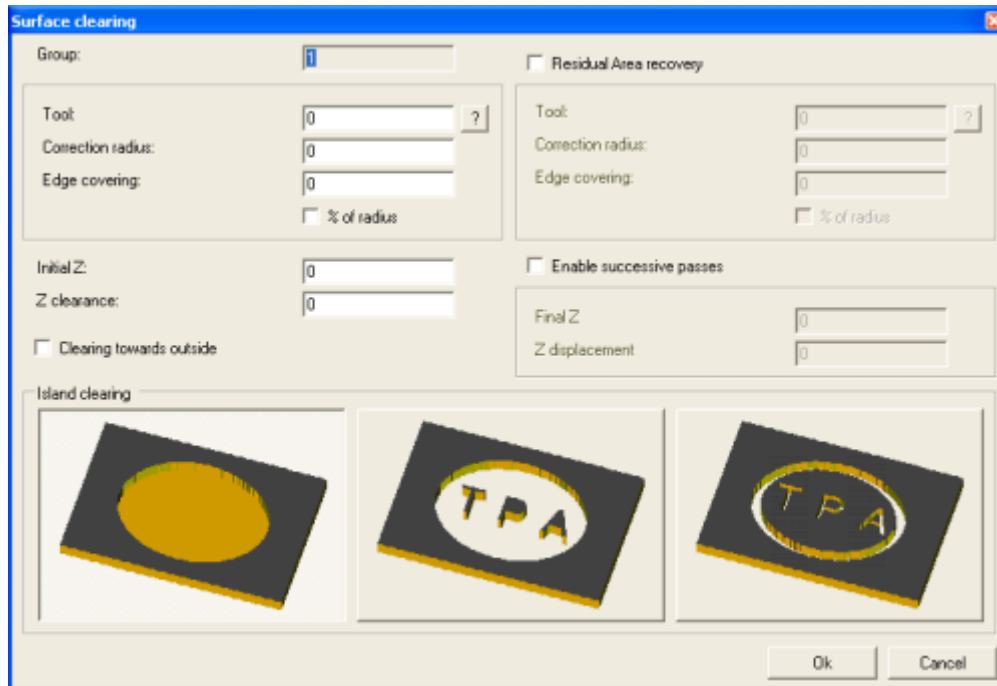
2.6.1 Surface Clearing Tool

The Surface Clearing operation is only enabled if at least one of the profile elements to be cleared is selected. The basic characteristic of the profile is that it must be a closed profile, that is, the final and initial points must coincide.

The clearing process is recalled by pressing the  button on the Tools toolbar or by means of the menu item "Tools" – "Clear a profile".

2.6.1.1 Setting Window for Surface Clearing Parameters

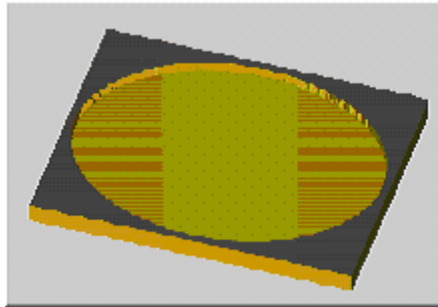
When the tool is selected the following clearing data setting window appears:



The fields in this window are:

- **Group:** Used for setting the machine group (group 2 is only enabled on machines with double beam)
- **Tool:** Used for setting the tool to be used for the clearing process; press the "?" button to select the tool directly from the window which lists all the tools available in the current outfits manager.
- **Correction radius:** Once the tool has been selected the value of the tool radius is assigned automatically; it is used to calculate the number of passes; it can be changed.
- **Edge covering:** Expresses, in mm, the amount by which the various passes must overlap; if the value is left as it is the clearing may not be complete (especially if the real diameter of the tool is a few tenths smaller than the indicated diameter). It is also possible to select the "radius %" field; in this case the value in mm is calculated from the percentage of the set radius.
- **Residual Area Recovery:** Enables the setting of a second tool (usually with a smaller diameter) which is used to clear any residual areas created after the clearing executed with the first tool. The meaning of the parameters of the second tool are the same as those listed above.
- **Initial Z:** is the final value of the clearing when there is only one pass; it is the value of the first pass if the clearing involves several passes.
- **Z Clearance:** is the Z value with which the movements in space are generated (in any case, the clearing process generates a single initial setup and possibly movements interpolated in space).
- **Enable successive passes:** Enables the clearing for several passes; therefore, "Final Z" and "Z Displacement" become relevant fields.
- **Final Z:** This value is important if the successive passes are enabled; it indicates the final clearing value.
- **Z Displacement:** This value is important if the successive passes are enabled; it indicates the displacement step between the various passes (to be programmed with a positive value even if the displacement is in the Z- direction).
- **Clearing towards outside:** Used to enable the clearing from the inside towards the outside; the default setting is from outside towards the inside.
- **Island clearing:** Three buttons are used to establish the clearing criteria. Three clearing modes are available for selection.

Mode 1:

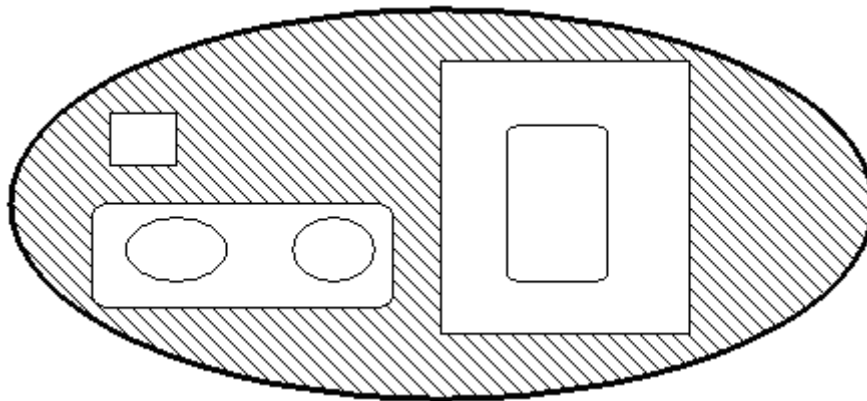


Simple clearing: ignores any internal profiles and simply applies the clearing inside the selected profile; the result is as follows:

Mode 2:



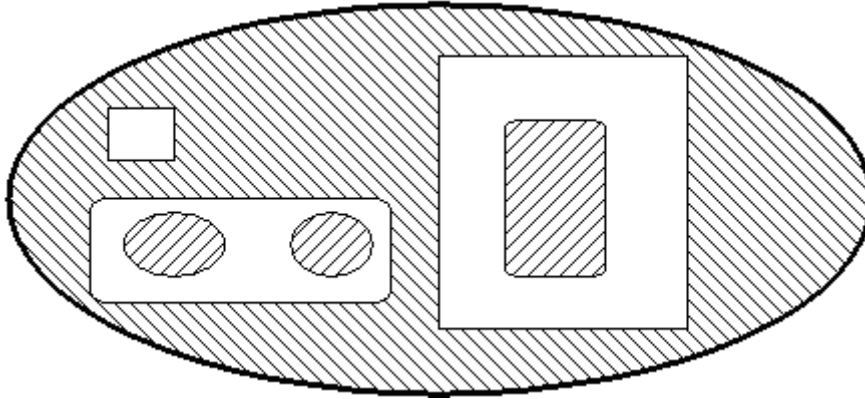
Clearing excluding the islands: clears the selected profile except for the (closed) profiles inside it; using the profiles of the previous example the result of the clearing of the ellipse is as follows:



Mode 3:




Alternating clearing: clears the selected profile and the profiles inside it alternately; using the previous example the result of the clearing is:



Up to 100 profiles, one inside the other, can be managed.

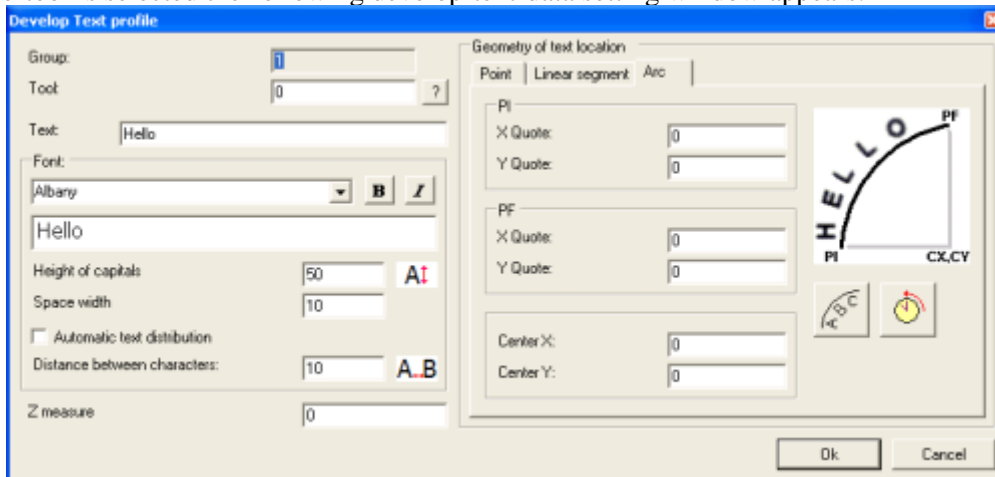
Clearing can be applied to all closed profiles, even those generated by macros (oval, ellipse, polygons, etc) or by tools (text).

2.6.2 Develop Test Tool

The Develop Test operation is recalled by pressing the  button on the Tools toolbar or by means of the menu item "Tools" – "Enter a Text". The item is only enabled with one selected face.

2.6.2.1 Setting Window for Text Parameters

When the tool is selected the following develop text data setting window appears:



When the tool is selected the following develop text data setting window appears:

- **Group:** Used for setting the machine group (group 2 is only enabled on machines with double beam)
- **Tool:** The tool to be used for writing the text is entered in this field ; press the "?" button to select the tool directly from the window which lists all the tools available in the current outfits manager.
- **Text:** The text to be transformed into profile is written in this field.
- **Font:** Select the font to be used to develop the text. A list of the True-Type (Scaleable) fonts installed in the system appear; the list can vary from one computer to the next depending on what has been installed (for example, Word and other applications add different fonts). Apply bold and

italic formats to the font by means of the "B" and "I" buttons.

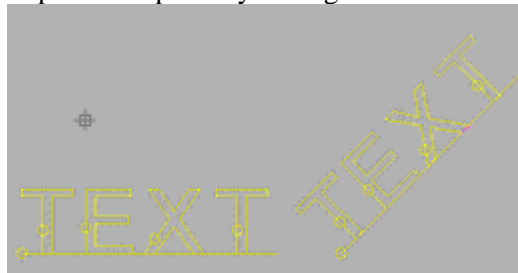
- **Height of capitals:** sets (in mm) the height of the character, using the capital A as reference.
- **Space width:** is the distance (in mm) of the "space" character.
- **Automatic text distribution:** used to distribute the text automatically on a hypothetical linear segment or a hypothetical arc; it has no relevance if the "Point" option is selected in the geometry of text distribution.
- **Distance between characters:** sets the distance (in mm) between one character and the next; this parameter is irrelevant if the automatic text distribution check box is selected.
- **Z measure:** sets the final depth of the character.

2.6.2.2 Geometry of text distribution

The text can be positioned using three different methods: Point, Linear Segment and Arc. For the Linear Segment and the Arc it is possible to specify if the distribution is to be done automatically with respect to the specified segment (linear or arc). In this case the text is distributed (calculating the space between the characters) so that it fills up the entire segment.

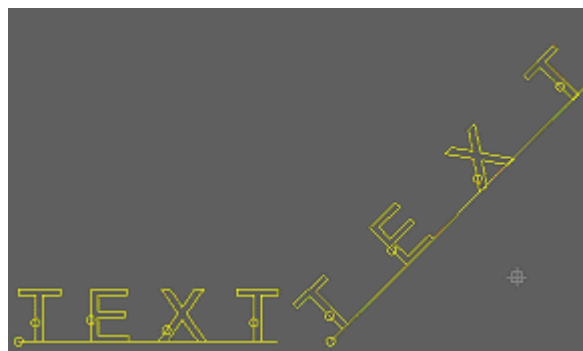
Point method

In this case specify an X and Y point and possibly an angle of rotation.



The example shows two linear texts starting from the indicated point and with the specified inclination (0 and 45 degrees).

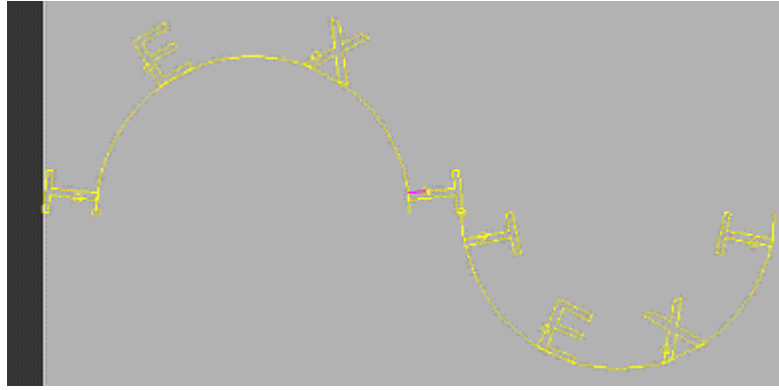
The second method is distribution along a Linear Segment; in this case specify the X and Y initial and final points of the linear segment.



The example shows two linear texts that start from the indicated point and end at the final point. In this example, the text is distributed along the specified linear segment by calculating the distance between one character and the next (option *Automatic text distribution*).

The third method is distribution along an Arc; in this case specify the X and Y initial and final points, the centre of the arc (absolute), the rotating direction of the arc (Clockwise or Counterclockwise) and if the text is to be positioned inside or outside the arc (outside by default)





The example shows two arcs, the first clockwise and the second counterclockwise, the first with the text outside the arc and the second with the text inside the arc. In this example, the text is distributed along the specified arc by calculating the distance between one character and the next (option *Automatic text distribution*).



Genesis Evolution

Il presente manuale è di proprietà della T.P.A S.p.a.
Ne è vietata la duplicazione non autorizzata.

La società si riserva il diritto di modificarne il contenuto in qualsiasi momento

T.P.A S.p.A Tecnologie e prodotti per l'automazione

Via Carducci 221
20099 Sesto S. Giovanni (MI)

Tel. 0236527550

Fax. 022481008

www.tpaspa.it