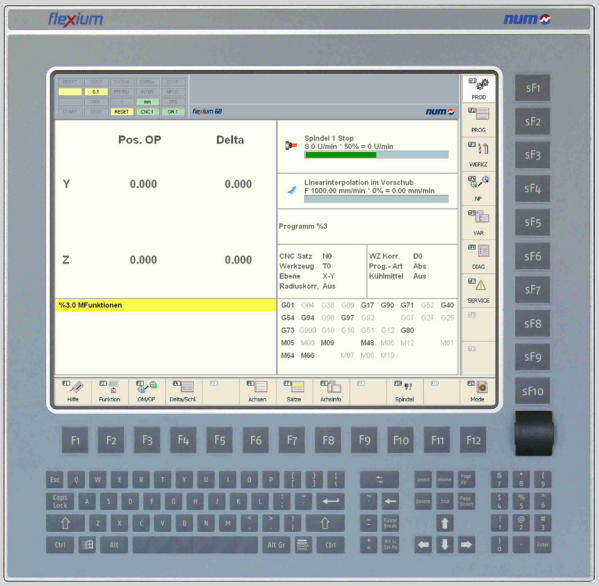
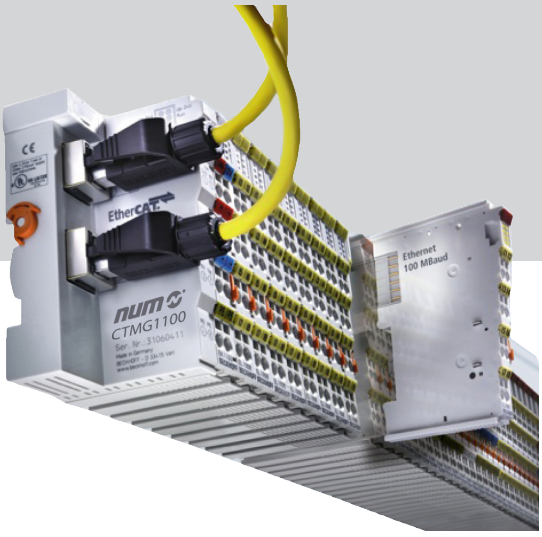
Flexium

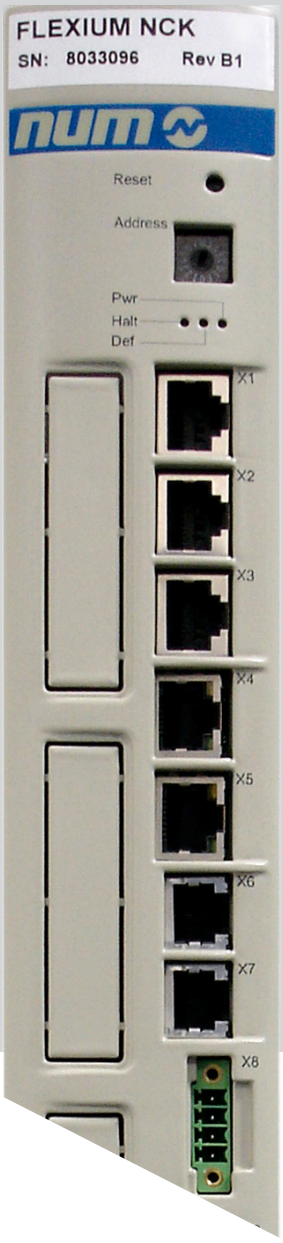
Programming manual

M00018EN-03

Edition 02.2013







www.num.com

num®
CNC HighEnd Applications

Despite the care taken in the preparation of this document, NUM cannot guarantee the accuracy of the information it contains and cannot be held responsible for any errors therein, nor for any damage which might result from the use or application of the document.

The physical, technical and functional characteristics of the hardware and software products and the services described in this document are subject to modification and cannot under any circumstances be regarded as contractual.

The programming examples described in this manual are intended for guidance only. They must be specially adapted before they can be used in programs with an industrial application, according to the automated system used and the safety levels required.

© Copyright NUM 2008
All rights reserved. No part of this manual may be copied or reproduced in any form or by any means whatsoever, including photographic or magnetic processes. The transcription on an electronic machine of all or part of the contents is forbidden.

© Copyright NUM 2008 software NUM Flexium family
This software is the property of NUM. Each memorized copy of this software sold confers upon the purchaser a non-exclusive licence strictly limited to the use of the said copy. No copy or other form of duplication of this product is authorized.

Summary

1 Foreword 13
1.1 Use of the manual 13
1.2 Use of symbols in the manual 13
1.3 Document revisions 14
1.4 How to read the manual 15
1.5 Use of this Programming Manual 16
1.5.1 Function Syntax Entry Conventions 16
1.5.2 NC Operating Modes 16
1.6 NUM Agency 17
1.7 Index 17

2 Review 21
2.1 System Overview 21
2.1.1 Overview of Modes 21
2.1.2 Defining a Program 21
2.1.3 Preparing a Program 22
2.2 Machine Overview 23
2.2.1 Review of Axis Definition and Direction 23
2.2.2 Machine coordinates 24
2.2.3 Definition of Travels and Origins 25
2.2.4 Part settings 27
2.2.4.1 Milling Machines Equipped with Rotary Tables 29
2.2.5 Tool Dimensions 31
2.2.5.1 Milling tools dimensions 31
2.2.5.2 Turning tools dimensions 32
2.2.6 Definition of Dynamic Tool Corrections 34
2.2.6.1 Dynamic Tool Corrections for Milling Tools 34
2.2.6.2 Dynamic Tool Corrections for Turning Tools 35

3 Structure of a Program 39
3.1 Word Format 40
3.1.1 General Word Format 40
3.1.2 Special Features for Dimensions 40
3.1.3 Internal Unit for Linear Axes 41
3.1.3.1 Correspondence between the word format and internal unit for linear axes 41
3.1.4 Internal Unit for Rotary Axes 41
3.1.4.1 Correspondence between the word format and the internal system unit for rotary axes 41
3.1.5 Examples of word formats 42
3.1.5.1 Word defining a dimension, address X (internal unit in µm) 42
3.1.5.2 Word defining a function, address G 42
3.2 Block Format 43
3.2.1 Examples of blocks 43
3.2.1.1 A block defining a tool change and calling up the tool correction 43
3.2.1.2 A block defining spindle rotation 44
3.2.1.3 A block defining a move 44
3.3 General Structure of a Program 45
3.3.1 General 45
3.3.2 Branches and Subroutine Calls 46
3.3.3 Program Numbering 47
3.3.3.1 Inhibiting display of subroutines being executed 47
3.4 Classification of Preparatory and Miscellaneous Functions 48
3.4.1 Classification of G Functions 48

3.4.1.1 Modal G Functions 48
3.4.1.2 Nonmodal G Functions 48
3.4.1.3 G Functions Incompatible with the state of the program 48
3.4.1.4 G Functions Associated with Arguments 49
3.4.2 Classification of M Functions 50
3.4.2.1 Modal M Functions 50
3.4.2.2 Nonmodal M Functions 50
3.4.2.3 «Pre» M Functions 50
3.4.2.4 «Post» M Functions 50
3.4.2.5 Decoded M Functions 51
3.4.2.6 Encoded M Functions 51

4 Axis Programming 55
4.1 General 55
4.2 Programming the Independent Secondary Axes 56
4.3 Programming of Carrier/Carried Parallel Axis Pairs 56
4.4 Programming Rotary Axes Modulo 360 Degrees 57
4.4.1 Rotary axis programmed in absolute dimensions (G90) 57
4.4.2 Rotary axis programmed in incremental dimensions (G91) 58
4.5 Programming Rotary Axes with Limited Travel 59
4.6 Programming of Axes A, B or C Declared as Nonrotary 59

5 ISO Programming 67
5.1 G Functions, M Functions and other Functions - Main summary tables 67
5.1.1 G Functions - Main summary table 67
5.1.2 M Functions - Main summary table 79
5.1.3 Other Functions - Main summary table 83
5.1.4 Word Format - Main summary table 86
5.2 Programming with Reference to Diameter or Radius 88
5.2.1 System Programmed with Reference to Diameter 88
5.2.2 System Programmed with Reference to Radius 88
5.3 Absolute or Incremental selection ► G90, G91 89
5.4 Plane Selection ► G17, G18, G19 91
5.5 Spindle Control 92
5.5.1 Spindle Direction of Rotation ► M03, M04, M05 92
5.5.2 Spindle Speed Control ► G97 94
5.5.2.1 Review of surface speed 96
5.5.2.2 Using a fixed spindle rotation speed (N) 96
5.5.2.3 Using a constant surface speed (V) 97
5.5.2.4 Spindle speed limit 97
5.5.3 Constant Surface Speed on Turning machine ► G96 98
5.5.4 Spindle Speed Limiting ► G92 S 99
5.5.5 Spindle Range ► M40, M41, M42, M43, M44, M45 101
5.5.6 Indexed Spindle Stop ► M19 102
5.5.7 Spindle Control Selection ► M64, M65, M62, M63, M61 103
5.5.8 Spindle Measurement Selection ► M66, M67, M68, M69 105
5.6 Rapid Positioning ► G00 106
5.6.1 Positioning with respect to the cutter dia. (R-/R+) example 108
5.7 Programming Movements 110
5.7.1 Linear Interpolation ► G01 110
5.7.1.1 Programming of additional axes and carrier/carried axis pairs 111
5.7.2 Positioning with respect to the cutter dia. (R+/R-) 113
5.7.3 Circular Interpolation ► G02, G03 114
5.7.3.1 Direction of movement according to the turret position with Turning Machine 116
5.7.3.2 Programming a circle by its radius (R..) 117
5.7.3.3 Programming a circle by the coordinates of its centre (I and J, plane G17) 117
5.7.3.4 Programming the additional axes and carrier/carried axis pairs 118
5.7.4 Helical Interpolation 119
5.7.5 Determination of the helix depth 120
5.7.5.1 Syntax according to the plane selected: G17/G18/G19 121
5.7.5.2 Programming of additional axes and carrier/carried axis pairs 121
5.7.6 Circular Interpolation Defined by Three Points ► G23 123

5.7.6.1 Syntax According to the Plane Selected: G17/G18/G19 124
5.7.7 Polar Programming 125
5.7.7.1 Polar Programming of a Line 126
5.7.7.2 Polar Programming of a Circle 128
5.7.7.3 Defining a Circle by the Arc Angle 132
5.7.8 Programming Fillets and Chamfers 138
5.7.8.1 Fillet between two interpolations ► EB+ 138
5.7.8.2 Chamfer Between Two Linear Interpolations ► EB- 139
5.7.9 Accurate End of Block Stop ► G09 141
5.8 «Inclined Axis» or «Inclined Wheel» 143
5.9 Initial Tool Positioning Before Machining on an Inclined Axis ► G07 145
5.10 Movement Along the Inclined Axis ► G05 146
5.10.1 Feed Rate 148
5.10.1.1 Feed Rate Expressed in Millimetres, Inches or Degrees per Minute ► G94 148
5.10.1.2 Programming additional axes 149
5.10.1.3 Programming of a modulo rotary axis programmed alone 149
5.10.1.4 Programming of modulo rotary axis and a linear axis 149
5.10.1.5 Programming of two modulo rotary axes 149
5.10.1.6 C axis feed rate for Turning Machine 150
5.10.2 Inverse Time Feed Rate Coding (V/D) ► G93 151
5.10.3 Feed Rate Expressed in Millimetres or Inches per Revolution ► G95 155
5.10.4 Tangential Feed Rate ► G92 R.. 157
5.10.5 Setting feed rate per axis set ► G92 F 160
5.10.6 Feed Rate Specific to Fillets EB+ and Chamfers EB- ► EF 161
5.11 Tool Programming 164
5.11.1 Tool Change ► M06 164
5.11.2 Tool Axis Orientation ► G16 167
5.11.3 Tool Offset activation ► D 170
5.11.3.1 Tool length correction (L) 171
5.11.3.2 Tool dimensions corrections 172
5.11.3.3 Tool radius corrections 172
5.11.3.4 Tool tip orientation (C) according to the turret position 174
5.11.3.5 Use of tool corrections (X, Z, R, C) on a lathe with two mechanically coupled turrets . 174
5.11.4 Tool radius compensation ► G41, G42, G40 176
5.11.4.1 Tool positioning 179
5.11.4.2 Radius correction according to the turret position 180
5.11.4.3 Tool outside the profile (line/line or circle/circle) 181
5.11.4.4 Tool outside the profile (line/circle) 183
5.11.4.5 Tool inside the profile 183
5.11.4.6 Tool inside the profile, special cases 184
5.11.5 Tool disengagement 184
5.11.6 Tridimentional tool correction with toroid or spherical tool ► G29 195
5.11.6.1 Concept of Surface and Normal Vector 197
5.11.6.2 Toroidal and Spherical Tools 198
5.11.6.3 Double Twist Machine Head 198
5.11.6.4 Geometric Transformations with 3-Axis Correction 199
5.11.6.5 Geometric Transformations with 5-Axis Correction 201
5.11.6.6 Tridimentional Tool Correction with Cylindrical Tool ► G43 202
5.12 Canned Cycles 203
5.12.1 Cycle Overview ► G8x 203
5.12.2 Cancellation of a Canned Cycle ► G80 205
5.12.3 Centre Drilling Cycle ► G81 206
5.12.4 Counterboring Cycle ► G82 208
5.12.5 Peck Drilling Cycle ► G83 210
5.12.6 Tapping Cycles ► G84 214
5.12.7 Rigid Tapping Cycle ► G84 217
5.12.8 Rigid Tap Removal ► G39+ 221
5.12.8.1 Execution of the Rigid Tap Removal Cycle 222
5.12.9 Reaming Cycle ► G85 223
5.12.10 Boring Cycle with Indexed Spindle Stop at the Bottom of the Hole ► G86 225
5.12.11 Drilling Cycle with Chip Breaking ► G87 227
5.12.12 Boring and Facing Cycle ► G88 231
5.12.13 Boring Cycle with Dwell at the Bottom of the Hole ► G89 233

5.12.14 Thread Chasing Cycle ► G31 238
5.12.14.1 Cycle repeat to achieve the thread precision 240
5.12.15 Cycle summary 241
5.12.16 Examples of G81 to G89 cycles 242
5.13 Turning specific canned cycles 245
5.13.1 Constant Pitch Thread Chasing Cycle ► G33 245
5.13.1.1 Tool retraction slope at the end the pass defined by R 247
5.13.1.2 Penetration defined by EB 248
5.13.1.3 Tapered thread cutting, angle defined by EA 248
5.13.1.4 Multiple start thread cutting 249
5.13.2 Sequenced Thread Cutting ► G38 251
5.13.3 Resume Thread Cutting ► G38+ 253
5.13.3.1 Inclusion of function G38+ in Cycle G33 254
5.13.3.2 Associating a Teach-in Cycle with Resume Thread Cutting Cycle G33 255
5.13.4 Rough Turning/Facing Cycle ► G64 256
5.13.4.1 Roughing cycle P or R 259
5.13.5 Groove Roughing Cycle ► G65 267
5.13.6 Plunging Cycle ► G66 272
5.13.7 Roughing Cycle with Groove ► G63 276
5.13.7.1 Definition of EB and EC 280
5.13.7.2 Cycle differences between turning and facing 281
5.14 Simple Pocket Cycle ► G45 289
5.14.1 Pass setting possibilities 291
5.14.2 Roughing cycle with a single tool 292
5.14.3 Roughing Cycle Steps 292
5.14.4 Roughing and finishing cycle with a single tool 293
5.14.5 Finishing cycle alone with a new tool 293
5.14.6 Pass setting in the plane (Q) 294
5.14.6.1 Pass setting in the tool axis (P) 294
5.15 Breaks in Sequence ► G77 297
5.15.1 Unconditional Branch to a Subroutine or Sequence of Blocks with Return 297
5.15.2 Subroutine Branch by M Function ► M 303
5.15.2.1 Specific Feature of a Call to a Subroutine Number Between %11001 to %11999 304
5.15.3 Branch to a Sequence without Return ► G79 305
5.15.4 Subroutine Call by PLC 307
5.15.5 Block Interrupt ► G10 308
5.15.5.1 Analysis of the interrupt 309
5.15.5.2 Special Use of Sequence Interrupt 311
5.15.6 Temporary Suspension of Next Block Preparation ► G79+/- 313
5.15.7 Emergency Retract ► G75 315
5.15.8 Branch to Automatic Homing Subroutine %9990 318
5.15.9 Subroutine Branch on a Reset %11000 319
5.15.10 Restrictions Related to Drip Feed Mode 320
5.15.11 Call to Subroutine Return Block ► G77 -i 321
5.15.12 ISO Program or Block Creation/Deletion ► G76+/- 322
5.15.13 Creating a Program ► G76+ H 323
5.15.14 Deleting a Program ► G76- H 324
5.15.15 Inserting a Block ► G76+ 325
5.15.16 Deleting a Block ► G76- 326
5.16 Movement Referential selection 327
5.16.1 Programming of Movements in Absolute Coordinates Referenced to the Measurement Origin ► G52 327
5.16.2 Datum Shift DAT1 and DAT2 Cancel/Enable ► G53, G54 330
5.16.3 Program Origin Preset ► G92 331
5.16.4 Program Origin Offset ► G59 333
5.16.4.1 Function G59 programmed in absolute dimensions (G90) 334
5.16.4.2 Function G59 programmed in incremental dimensions (G91) 334
5.16.4.3 Features related to arguments I, J (K) 334
5.16.5 Angular offset ► ED.. 342
5.16.6 Table Offset by DAT3 with Milling Machine ► DAT3 346
5.17 Spline Curve Interpolation ► G48, G06, G49 348
5.17.1 Spline Curve Interpolation ► G48 349

5.17.2 Spline Curve Execution Command ► G06 351
5.17.3 Freeing Memory by Deleting a Spline Curve ► G49 356
5.18 Coordinates Systems with C Axis with Turning Machine 357
5.18.1 Programming in Polar Coordinates ► G20 357
5.18.2 Definition of the Start X for Interpolation on the C Axis ► G98 358
5.18.3 Programming in Cartesian Coordinates ► G21 359
5.18.4 Programming in Cylindrical Coordinates ► G22 364
5.19 Other Functions 367
5.19.1 Dwell ► G04 367
5.19.2 Programmed Feed Stop ► M12 368
5.19.3 Feed Increase ► G12 369
5.19.4 Programming in Inches or Metric Data ► G70, G71 370
5.19.5 Axis Clamping and Unclamping ► M10, M11 372
5.19.6 Coolant ► M08, M07, M09 373
5.19.7 Program Stop ► M00 374
5.19.8 Optional Stop ► M01 375
5.19.9 Inhibition of MDI Modes ► M999, M998 376
5.19.10 Forced Block consecutive execution ► M997 378
5.19.11 Potentiometer Inhibit ► M49, M48 379
5.19.12 Block Skip ► / N 380
5.19.13 Acceleration Reduction ► EG 381
5.19.14 Scaling Factor ► G74, G73 382
5.19.15 Mirror function ► G51 386
5.19.15.1 Influence of the mirror function on the A, B or C rotary axes 387
5.19.15.2 Influence of the mirror function on circular interpolations 387
5.19.16 Processing of Blocks and Programmed G and M Functions ► G999, G998, G997 389
5.19.17 Tridimentional curve smoothing ► G104 391
5.20 Message Transmission ► $0 393
5.20.1 Message Transmission to the Display 393
5.21 Message Display with request for an Operator Response ► L../E..,$ 395
5.22 Display of Messages with Parametric Value ► $ 397
5.23 Transfer of the Current Values of L Variables and E Parameters into the Part Program ► G76 398
5.24 Special Programming for Multiple Channels 400
5.24.1 Program Declaration 400
5.24.2 Programming Notes 400
5.24.3 Programming the Spindles 403
5.24.4 Releasing the Current Spindle in the Channel ► M61 404
5.24.5 Channel synchronisation with markers ► G78 405
5.24.5.1 Programming a Meeting Point 407
5.24.5.2 Programming by function G78 407
5.24.5.3 Programming with M functions 408
5.25 Special Programming of Auxiliary channels 409
5.25.1 Program Declaration and Storage 409
5.25.2 Programming of the Auxiliary channels 410
5.25.2.1 Emergency Retraction on an Auxiliary Channel 410
5.25.2.2 Exchanging Axes between Channels 411
5.25.2.3 Exchanging Spindles Between Channels 411
5.26 Special Features of Mixed Machines (MX) 412
5.26.1 Features Related to the Machine Axes 412
5.26.1.1 Boring Machine 413
5.26.1.2 Mixed Type Machine 413
5.26.2 Assignment of Tool Dimensions to the Axes 414
5.26.3 Preselection of the Program Origin (with function G92) 416
5.26.4 Notes on Programming the ISO Functions 417

6 Profile Geometry Programming 421
6.1 Profile Geometry 421
6.1.1 Definition of the Addresses Characterising PGP 422
6.1.1.1 Addresses with Values ► X../Y.. , Z.., EA, X.. Y.., I.. , K.., J.., R.., EB+, EB- 422
6.1.1.2 Addresses without Values ► ET, ES, E+/E- 425
6.1.2 Programming of Geometric Elements 427

6.1.2.1 Programming of Fully Defined Geometric Elements 427
6.1.2.2 Programming of Geometric Elements not Completely Defined....429
6.1.3 Programming Chamfers and Fillets Between Two Elements 443

7 Parametric Programming 455
7.1 "L" Local variables.... 455
7.1.1 Arithmetic Operations 456
7.1.2 Arithmetic Functions 456
7.1.3 Logic Operations.... 456
7.1.4 Comparison Symbols Used with L Variables....457
7.1.5 Conversion of the Internal Unit 457
7.2 Programming Syntax for L Variables 458
7.2.1 Assignment of a Variable to an NC function 458
7.2.2 Declaration of a Variable in the Program....459
7.2.3 Test of a Variable for a Conditional Branch 460
7.2.4 Notes on Programming Variables L100 to L199 and L900 to L959 461
7.2.5 Equivalence of Variables L900 to L925 461
7.2.6 Symbolic Addressing of Variables L900 to L925 and L926 to L951....461
7.3 Address Equivalences 470

8 External E Parameters 475
8.1 E Parameters - Main summary table....475
8.2 E Parameters description 480
8.2.1 Arithmetic Operations 481
8.2.2 Arithmetic Functions 481
8.2.3 Logic Operations....481
8.2.4 Comparison Symbols Used with External E Parameters 482
8.2.5 Conversion of the Internal to Program Unit 482
8.2.5.1 Examples 482
8.3 External E Parameter Programming Syntax 483
8.3.1 Assignment of an External Parameter to an NC Function.... 483
8.3.2 Declaration of an External Parameter in the Program.... 484
8.3.3 Test on an External Parameter for a Conditional Branch 485
8.4 How to read an E parameters table description 486
8.5 List of External E Parameters.... 488
8.5.1 Parameter for exchange with the machine application (PLC).... 488
8.5.2 Program Analysis Access Parameters 491
8.5.3 NC Configuration access parameters 498
8.5.4 Input / Outputs 501
8.5.5 Machining Parameters....503
8.5.6 Channel Axis Access Parameters 509
8.5.7 Spindle Access Parameters....516
8.5.8 Common NC Parameters 519
8.5.9 Machine Axis Access Parameters....521
8.5.10 Axis adjustment 527
8.5.11 Special purpose parameters....529
8.5.11.1 Delayed power down on machine errors 529
8.5.11.2 Axis / Spindle switching parameters.... 530
8.5.11.3 Spindle synchronisation....531
8.5.11.4 Drive access parameters.... 533

9 List of Errors 539
9.1 Miscellaneous Errors and Machine Errors 539
9.2 Parametric Programming Errors....542
9.3 Profile Geometry Programming (PGP) Errors 543
9.3.1 The end point cannot be calculated from the elements in the blocks 543
9.3.2 An intermediate point cannot be calculated from the data in two blocks 543
9.3.3 An intermediate point cannot be calculated from the data in three blocks.... 543
9.3.4 Fillet or chamfer definition errors 543
9.3.5 Miscellaneous errors in PGP 543
9.4 Miscellaneous Errors 544
9.5 Request for Movements Outside the Machine Travel Limits 544

9.6 Structured Programming Errors 544
9.7 Axis Errors 545
9.8 CNC - PLC Communication 545
9.9 System options errors 545
9.10 Axes Not Identified on the Bus 545
9.11 Dynamic Operators in C 546
9.12 Spline Curve Interpolation Errors 546
9.13 Cycle Programming Errors 547

Page deliberately empty

Summary

1 Foreword 13
1.1 Use of the manual 13
1.2 Use of symbols in the manual 13
1.3 Document revisions 14
1.4 How to read the manual 15
1.5 Use of this Programming Manual 16
1.5.1 Function Syntax Entry Conventions 16
1.5.2 NC Operating Modes 16
1.6 NUM Agency 17
1.7 Index 17

Page deliberately empty

1 Foreword

1.1 Use of the manual

This manual is for use by qualified personnel only.

To reduce the risk due to an improper use of the product, do not perform any servicing other than that contained in this manual unless you are qualified to do so.

To ensure proper use of these products, please always read this User's manual carefully and retain for future reference. Should the unit require maintenance, please contact a NUM authorized service center only.

1.2 Use of symbols in the manual

Symbol	Meaning of the symbol
	Important notes on product use.
	You may find this symbol whenever a wrong operation may damage the product.
	You may find this symbol whenever a dedicated Milling section or chapter is used.
	You may find this symbol whenever a dedicated Turning section or chapter is used.
	Please note that whenever functions in the manual present this Milling / Turning symbol means that such sections or chapters are used upon both cases.
xxxxxxx	Linked key word. (Blue highlight words)

1.3 Document revisions

Date	Index	Reason for revision	See chapter
12 - 2008	00	Document creation	-
05 - 2010	01	Internal revision "All-in-one Programming Manual" creation	-
04 - 2011	02	- New chapter 8 dedicated to the E parameters description and use. - Chapter 9 Error list updated.	-
02 - 2013	03	Chapter 1: - New Note upon the use of G76+. - New G78 description. - New function G153 to G159 for the measure cycles. - New function G199 "Programmed homing". Chapter 2: - New informations upon the Definition of Dynamic Tool Corrections. Chapter 3: - New informations upon the Encoded M Functions. Chapter 5: - New informations upon the use of G23. - New informations upon the use of Subroutine Branch by M Function. - New informations upon the use of Emergency Retract G75. - New Remark upon Function G59 programmed in incremental dimensions. - New informations upon the use G21 Feed rate limiting. - New informations upon the use of G70, G71. - New Note upon Subroutine Call by M Function. - New informations and Note upon the use of G78. - New informations upon Emergency Retraction on an Auxiliary Channel. Chapter 8: - New E parameters: E11022, E20100 and E20101, E41010, E90224 to E90227, E930xx, E971xx, E98600 to E98663, E942xx, E943xx, E34200, E34201, E34300, E34301. - Exxxx parametes upgraded: E10000 to E10031, E20000 to E20031, E30000 to E30255, E33000 to E33999, E37000 to E37127, E40000 to E40255, E43000 to E43999, E46000 to E46999, E47000 to E47127, E41102, E34000 and E34001, E34100 and E34101, E44000 and E44001, E44100 to E44103, E53xxx, E54xxx, E79003, E8x000 to E8x999, E952xx, E90120 to E90123, E35100 to E35131. Chapter 9: - Error 78 new informations. - Error 10 to 14 deleted. - New Error 253. Changed the position of the G, M Functions summary table. Changed the position of the E Parameters summary table.	 1.6 2.2.6 3.4.2.6 5.6.6 5.14.2 5.14.7 5.15.4.2 5.17.3 5.18.4 5.23.2 5.23.5 5.24.2.1 8.4 9.1 9.1 9.9 5.1 8.1

1.4 How to read the manual

The machine Flexium Programming Manual gives information on how to set the parameters to customize the Flexium for a particular Milling and Turning machine-tool.

Chapter	Description
1	**Foreword** This chapter introduces: - The use of the Programming manual. - Function Syntax Entry Conventions.
2	**Review** This chapter introduces: - General description of the NC and its use with the machine tool. - Review of the rules and standards related to the NC/machine-tool combination.
3	**Structure of a Program** This chapter introduces: - Rules for writing a part program by assembling characters into words. - Words into blocks and blocks into a complete program.
4	**Axes Programming** This chapter introduces: - Description of the features related to axes programming.
5	**ISO Programming** This chapter introduces: - Detailed description of functions related to ISO programming for Milling and Turning machine.
6	**Profile Geometry Programming** This chapter introduces: - Detailed description of profile geometry programming (PGP), - Using PGP used to define contours as a sequence of geometric elements, with computation of intermediate points. PGP is an extension of ISO programming.
7	**Parametric Programming** This chapter introduces: - Gives the possibility of assigning variables to NC functions. The values of the variables can be obtained by computation or by reading machine data.
8	**External E parameters** This chapter introduces the detailed information upon description and use of the E parameters.
9	**List of NC error numbers and definitions** This chapter introduces the NC error list.

1.5 Use of this Programming Manual

1.5.1 Function Syntax Entry Conventions

The lines (blocks) of a part program include several functions and arguments.

Special syntax rules apply to each of the functions described herein. These syntax rules specify how the program blocks must be written.

Certain syntax formats are given as a line. The following conventions simplify writing the line:

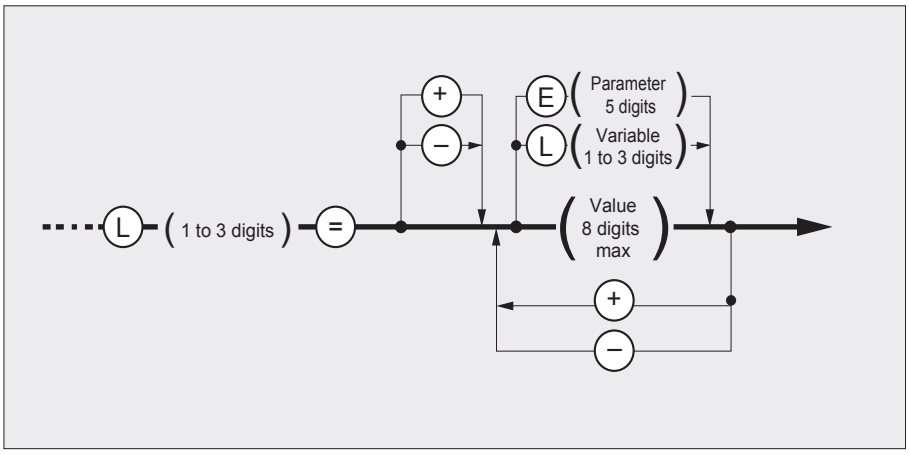
- the function to which the syntax format is related is highlighted by boldface type
- terms between square brackets «**[..]**» are optional functions or arguments in the block (except for symbolic programming where square brackets are part of the syntax)
- «**/**» indicates a choice between several terms (equivalent to «or») (except in parametric programming where "/" means divide)
- «**..**» after a letter replaces a numerical value
- «**...**» replaces a character string (for instance a message).

Examples

Syntax of function G12

N.. [G01/G02/G03] **G12** X.. Y.. Z.. [F..] [$0…]

Syntax in the form of a Conway diagram



1.5.2 NC Operating Modes

Certain NC operating modes are mentioned herein when they are directly related to the use of ISO functions. Additional information can be found in the Flexium HMI help.

1.6 NUM Agency

The list of NUM agencies is given click here http://www.num.com

1.7 Index

The general index at the end of the manual facilitates access to information by keywords.
Please click here index

Page deliberately empty

Summary

2 Review 21
2.1 System Overview 21
2.1.1 Overview of Modes 21
2.1.2 Defining a Program 21
2.1.3 Preparing a Program 22
2.2 Machine Overview 23
2.2.1 Review of Axis Definition and Direction 23
2.2.2 Machine coordinates 24
2.2.3 Definition of Travels and Origins 25
2.2.4 Part settings 27
2.2.4.1 Milling Machines Equipped with Rotary Tables 29
2.2.5 Tool Dimensions 31
2.2.5.1 Milling tools dimensions 31
2.2.5.2 Turning tools dimensions 32
2.2.6 Definition of Dynamic Tool Corrections 34
2.2.6.1 Dynamic Tool Corrections for Milling Tools 34
2.2.6.2 Dynamic Tool Corrections for Turning Tools 35

Page deliberately empty

2 Review

The aim of this chapter is to introduce concepts that will be detailed in the rest of the manual, rather than to reflect the way an operator works on the machine.

For instance, in the chapter 2.2.4 (Part settings), the aim is to define the offsets and corresponding origins or zero points rather than give a method for measuring the offsets.

This manual covers Turning, Milling and Special machines instructions. When a function is available only on a certain type of machine, it will be indicated precisely.

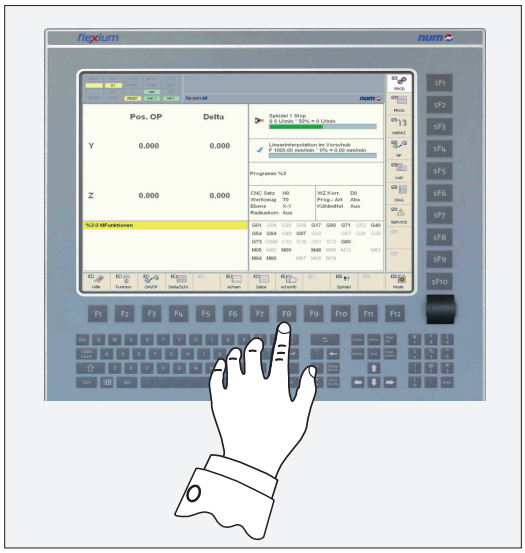
2.1 System Overview

2.1.1 Overview of Modes

The operator uses the numerical control (NC) in various operating modes accessible from the operator panel.

Each mode corresponds to a particular use of the numerical control (continuous machining, program loading, tool setting, etc.).

For additional information about Modes, please refer to the Flexium HMI on line help.



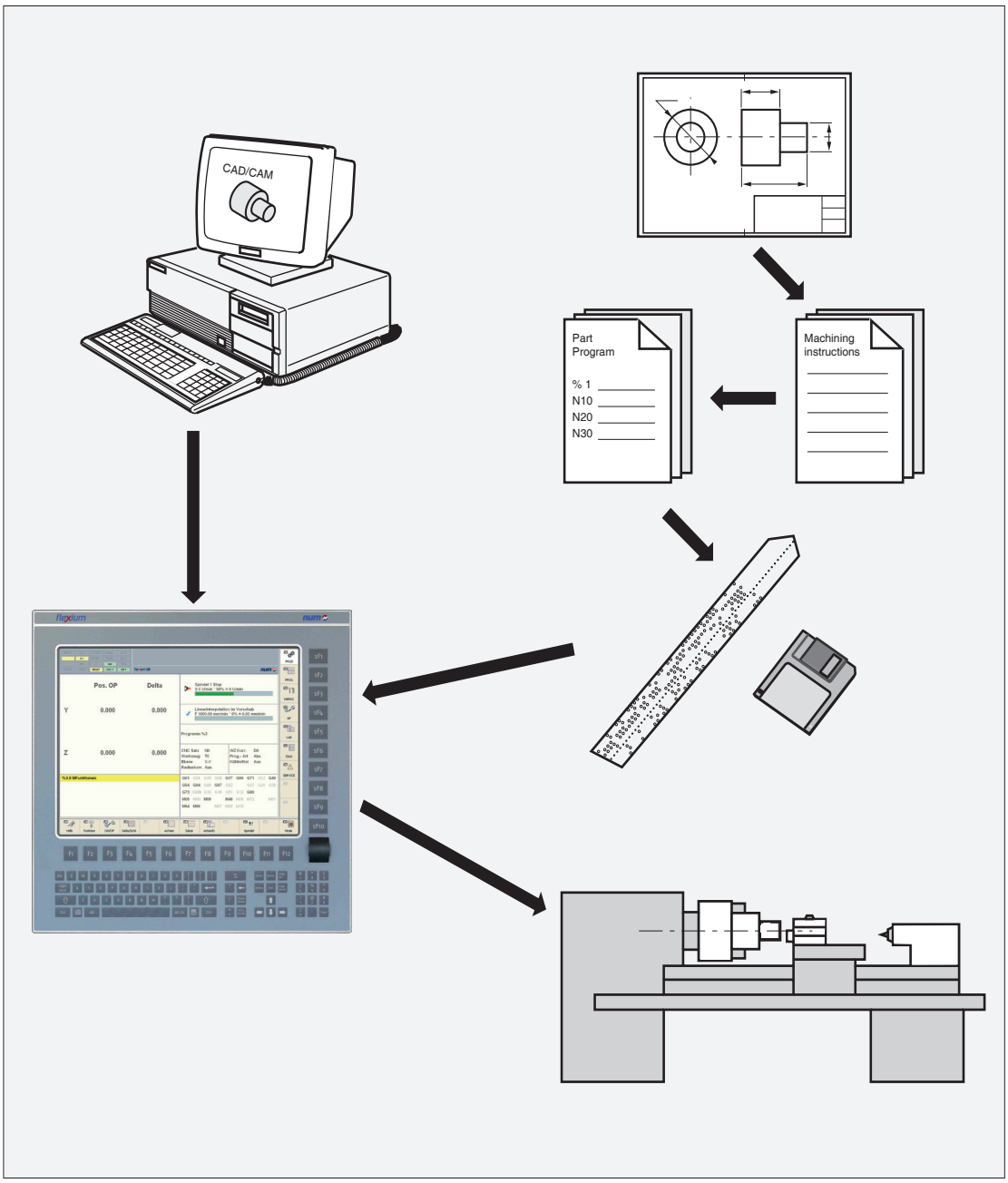
2.1.2 Defining a Program

A program is a sequence of instructions written in a programming language specific to the numerical control (the most widely used is ISO code: International Standards Organization).

The numerical control interprets the program to control actions on the machine.

2.1.3 Preparing a Program

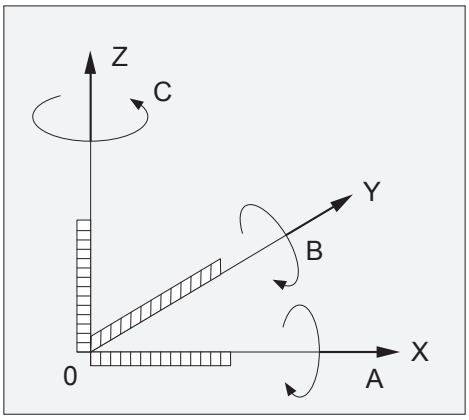
A part program can be created directly on the machine, on a remote editor or using a CAD/CAM system.



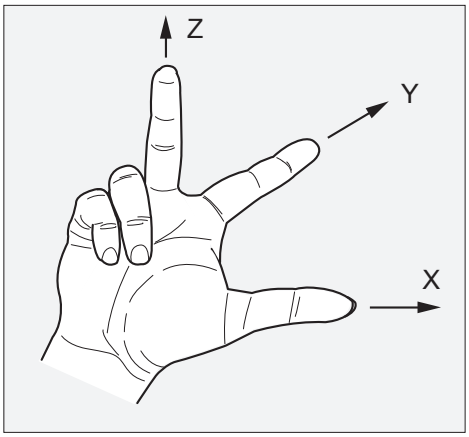
2.2 Machine Overview

2.2.1 Review of Axis Definition and Direction

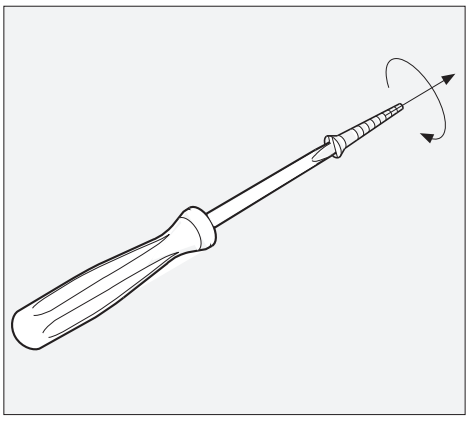
A coordinate system is used to identify the positions and movements of an object with respect to an origin or zero point.
A rectangular cartesian coordinate system is a direct three-axis system of three linear axes, X, Y and Z, with which are associated three rotary axes, A, B and C.



The direction of axes X, Y and Z is easily remembered by the right-hand rule.



The positive direction of rotation of a rotary axis corresponds to the direction of screwing of a right-hand screw on the associated axis.



2.2.2 Machine coordinates

The manufacturer defines the coordinate system associated with the machine in accordance with standard ISO 841 (or NF Z68-020).

The coordinate system measures tool movements with respect to the part to be machined, assumed fixed.

When it is the part that moves, it may be more convenient to identify its movements. In this case, axes X', Y' and Z', pointing in opposite directions from axes X, Y and Z, are used.

The direction of the axes of a machine depends on the type of machine and the layout of its components.

For a Milling Machine:
- the Z axis is the axis of the main spindle when this axis is parallel to one of the slideways,
- positive movement along the Z axis increases the distance between the part and tool,
- the X axis is perpendicular to the Z axis and corresponds to the largest excursion,
- the Y axis is perpendicular to the X and Z axes.

Rotary axes A, B and C define rotations around axes parallel to X, Y and Z.

Secondary linear axes U, V and W may or may not be parallel to primary axes X, Y and Z.

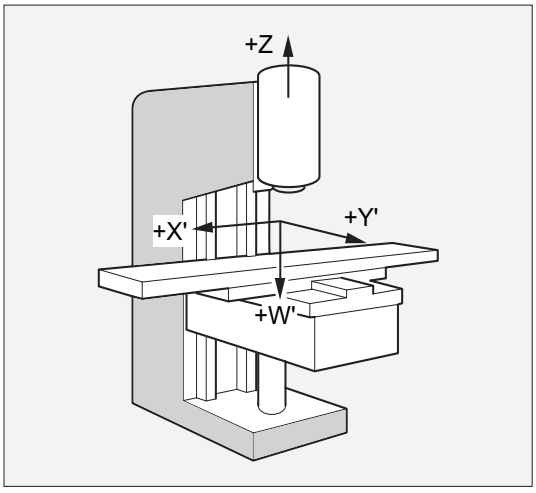
For a Turning Machine:
- the Z axis is the same as the spindle axis,
- the X axis is perpendicular to the Z axis and corresponds to radial movement of the tool-holder turret,
- the Y axis (generally a dummy axis) forms a right-handed coordinate system with the X and axes.

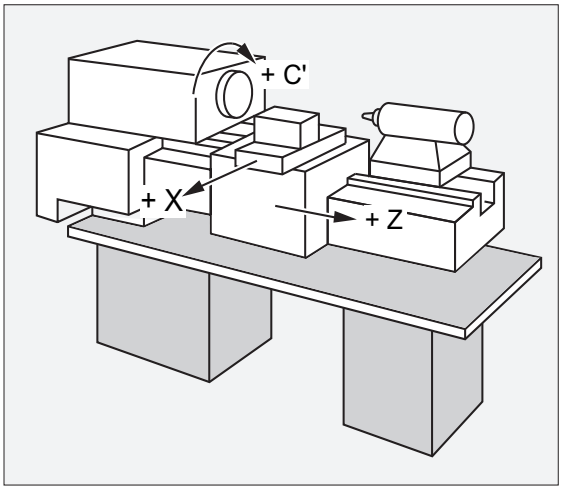
Positive movement along the Z or X axis increases the distance between the part and the tool.

Rotary axes A, B and C define rotations around axes parallel to X, Y and Z.

Secondary linear axes U, V and W may or may not be parallel to primary axes X, Y and Z.

For more details, refer to the above-mentioned standard.





2.2.3 Definition of Travels and Origins

All programmed movements are eventually related to the Machine Origin ('OM' in the rest of this document). The machine Origin is defined in relation with the Measure Origin ('Om' in the rest of this document). More and more machines are equipped with absolute measurement devices allowing the Measure Origin (Om) to be automatically determined at power up.

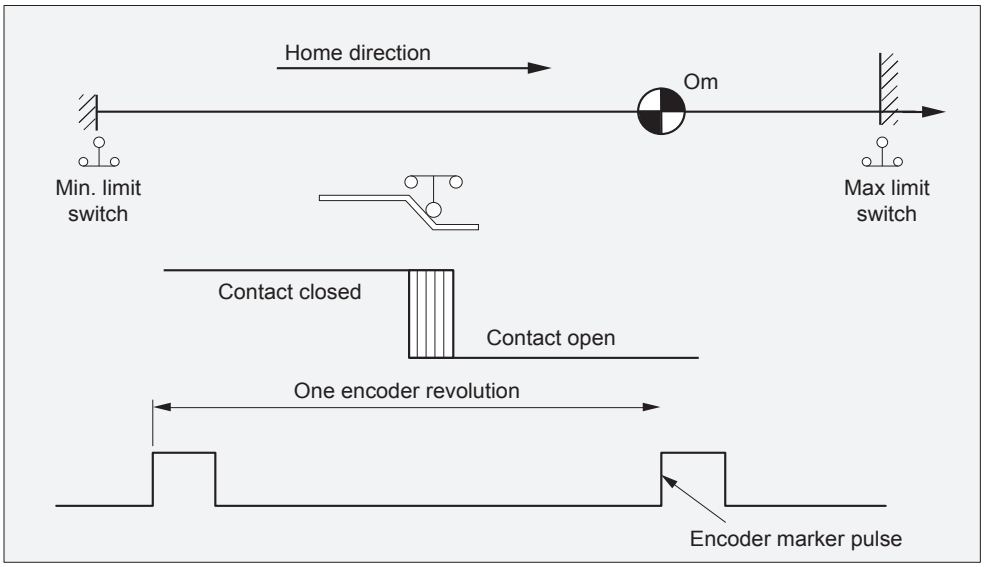
For the machines not equipped with such devices an Homing procedure is mandatory in order for the Control System to know the axes position. Once this is completed the axes travel will be automatically limited to the values defined by the machine manufacturer.

Symbols

Om :	Measure Origin: reference of the measurement system. It is evaluated either at the completion of the homing procedure or automatically at power up for absolute measurements systems.
OM :	Machine Origin: reference for all programmed movements.

The homing procedure is complete for each of the axes when:

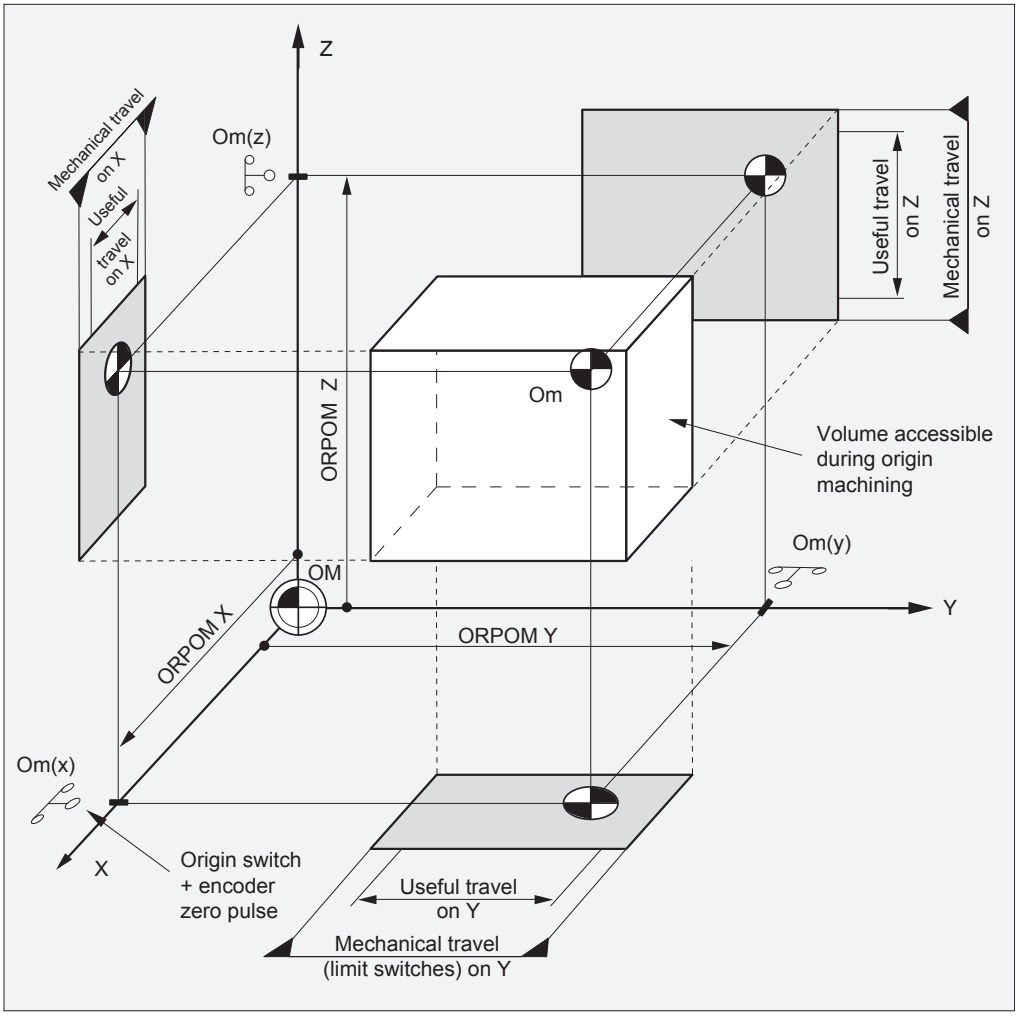
- Automatically if the machine is equipped with an absolute measurement system.
- After a short displacement (~40mm) in Home mode for the machines equipped with semi absolute measurement systems (Coded reference scales).
- After reaching the first marker pulse following the activation of the home switch for machine equipped with non absolute measurement systems.

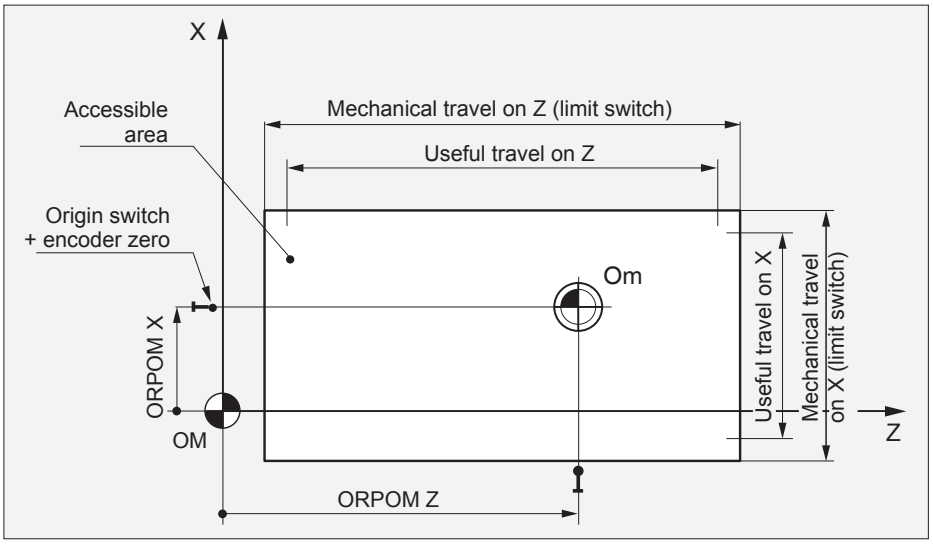


When homing is completed, the system applies the offset defined by the manufacturer to the relevant axis in order to establish the machine origin (OM).
This parameter is often named ORPOM.

Machine origin offset (Om/OM) = ORPOM

The useful travel on each axis is limited by software limits whose values are defined by the manufacturer.





2.2.4 Part settings

The part program is established in reference to a Program Origin (OP).

However, the part is mounted on the machine in reference to a Part Origin (Op).

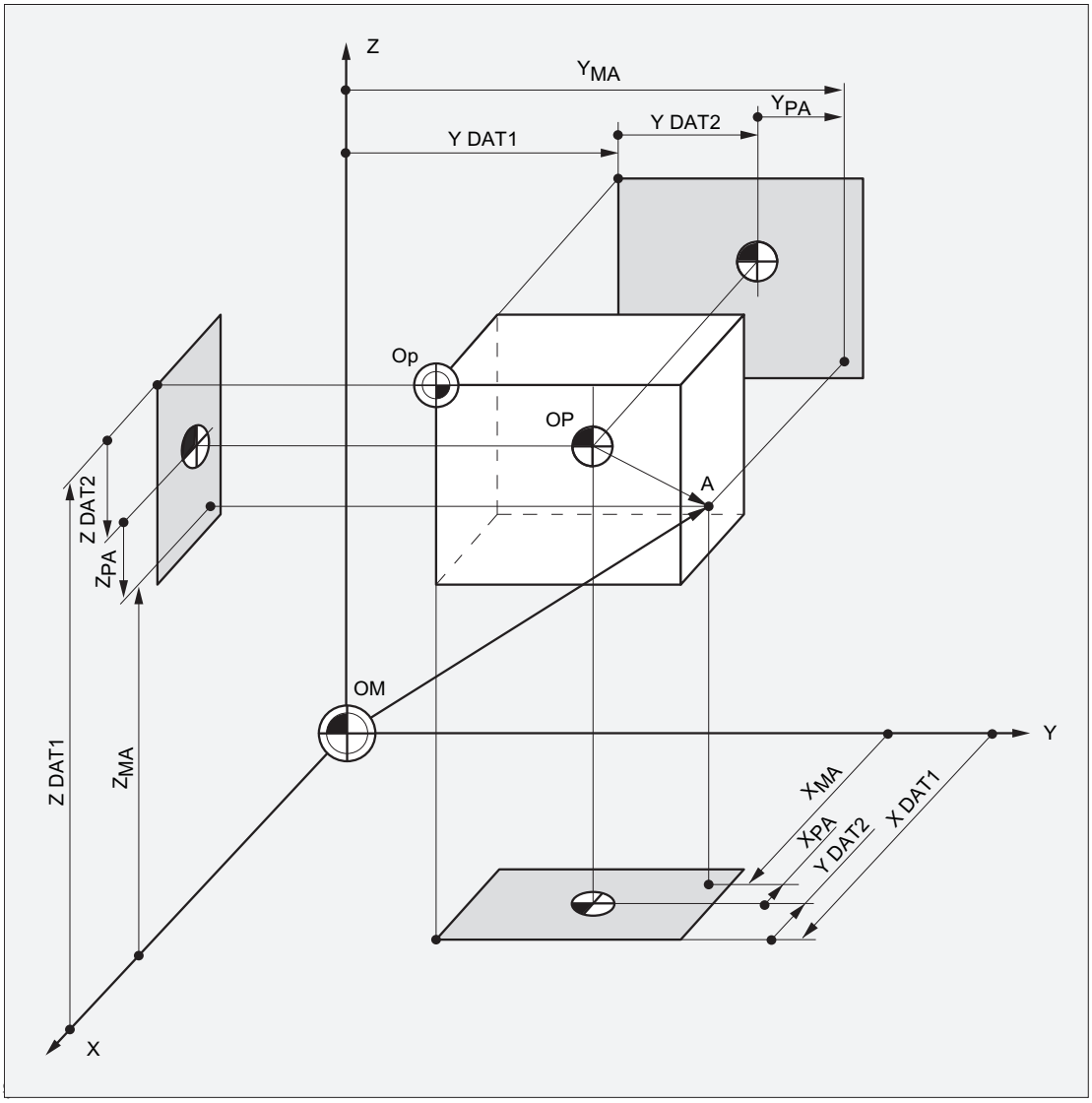
OP :	Program Origine : Defined from Op for each axis via parameter DAT2:
Op :	Part Origine : Defined from OM for each axis via DAT1.

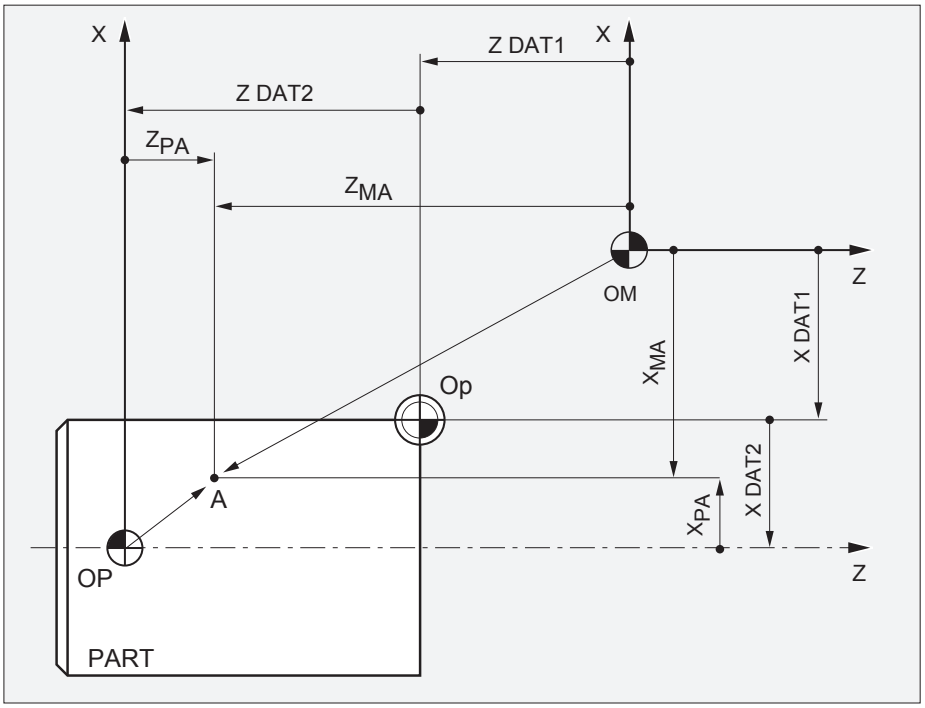
Program origin offset (OP/Op) = DAT2

Part origin offset (Op/OM) = DAT1

It is possible to define via programming the DAT1, DAT2 values for each axis.

The coordinates of a point (A) defined with respect to the program origin (OP) are converted by the NC to coordinates with respect to the machine origin (OM):





Program dimensions (with respect to OP)	Measurement dimensions (with respect to OM)
X_{PA}	$X_{MA} = X_{PA} + X\ DAT1 + X\ DAT2$
Y_{PA}	$Y_{MA} = Y_{PA} + Y\ DAT1 + Y\ DAT2$
Z_{PA}	$Z_{MA} = Z_{PA} + Z\ DAT1 + Z\ DAT2$

Programmed shifts can be added to the program dimensions.

2.2.4.1 Milling Machines Equipped with Rotary Tables

When a part is mounted on a rotary table, rotating this table will result in a modification of the part origin and therefore the program origin.

This modification is made along the two axes perpendicular to the axis of rotation

An additional shift (DAT3) is created to take this modification into account.

Considering OC as the center of rotation we have the relations:

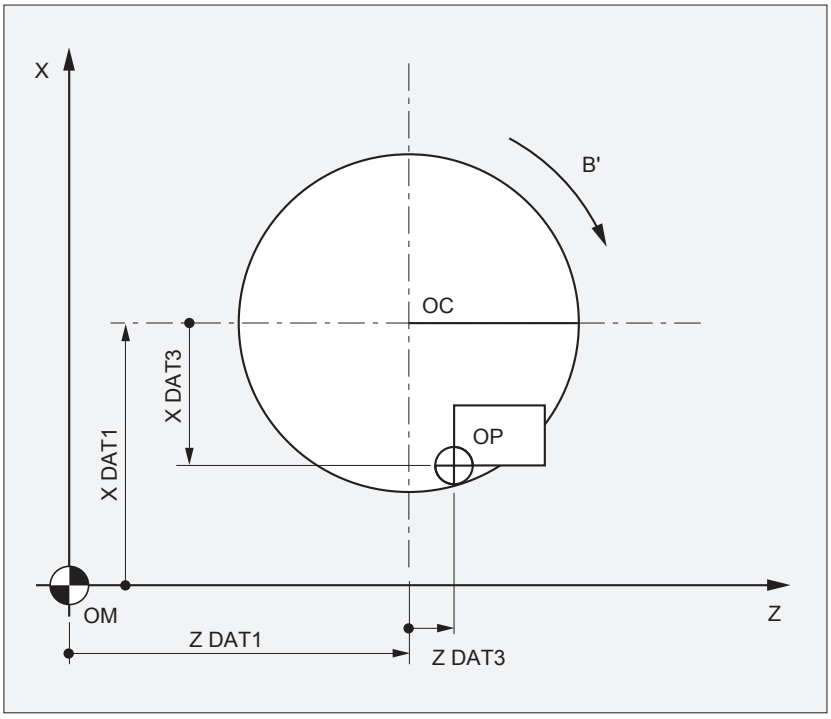
Offset to the centre of rotation (OC/OM) = DAT1

Offset to the part origin (OP/OC) = DAT3

For axes other than those affected by rotation, only DAT1 and DAT2 apply.

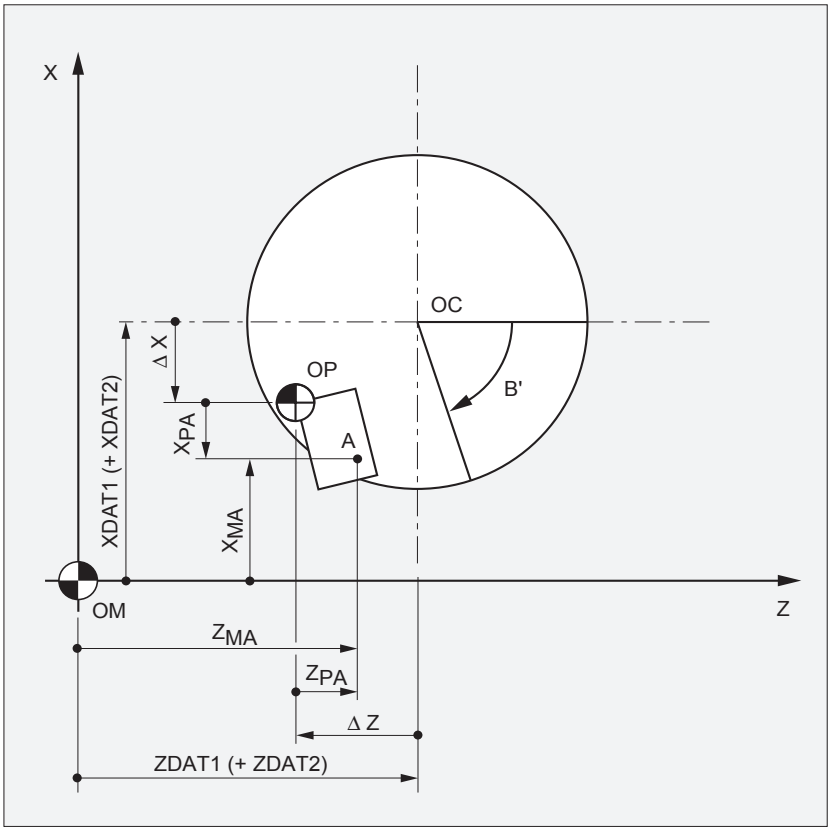
Example: rotary axis B

Rotation is around an axis parallel to the Y axis. Therefore, the axes affected by the rotation are Z and X. For the B axis at Zero/OP DAT3 will be defined as follows:



After a rotation, The NC will apply the corrections ΔX and Δ Z so that the position A is correct.

Remark. The correction will only be taken into account when X and/or Z are programmed. There is no continuous compensation during the table rotation:



Program dimensions (with respect to OP)	Measurement dimensions (with respect to OM)
X_{PA}	$X_{MA} = X_{PA}$ + XDAT1 (+ XDAT2) + ΔX with ΔX = XDAT3 x cos B - ZDAT3 x sin B
Y_{PA}	$Y_{MA} = Y_{PA}$ + YDAT1 + YDAT2
Z_{PA}	$Z_{MA} = Z_{PA}$ + ZDAT1 (+ ZDAT2) + ΔZ with ΔZ = ZDAT3 x cos B + XDAT3 x sin B

2.2.5 Tool Dimensions

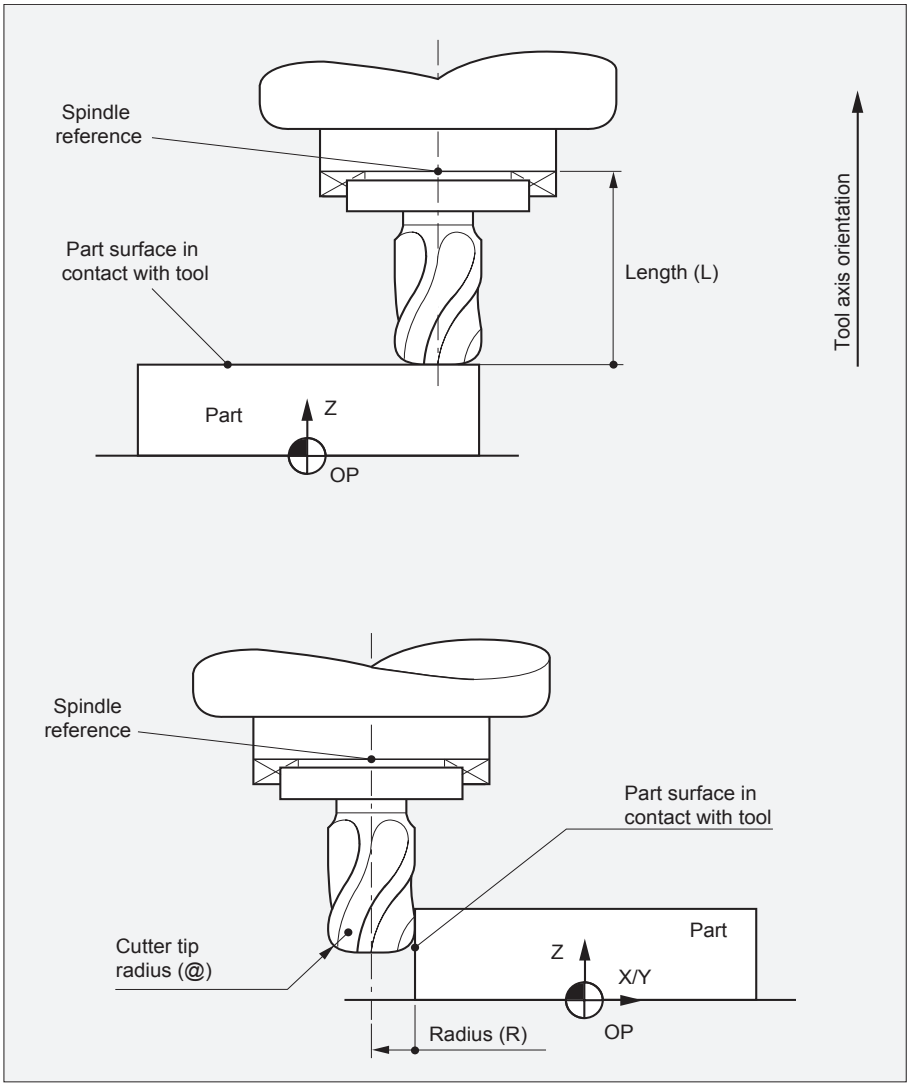
Flexium CNC considers two types of tools :

- Milling tools
- Turning tools

According to the tool type, the tool offset data are different.

2.2.5.1 Milling tools dimensions

Tool dimension = distance from tool cutting edge to spindle reference point

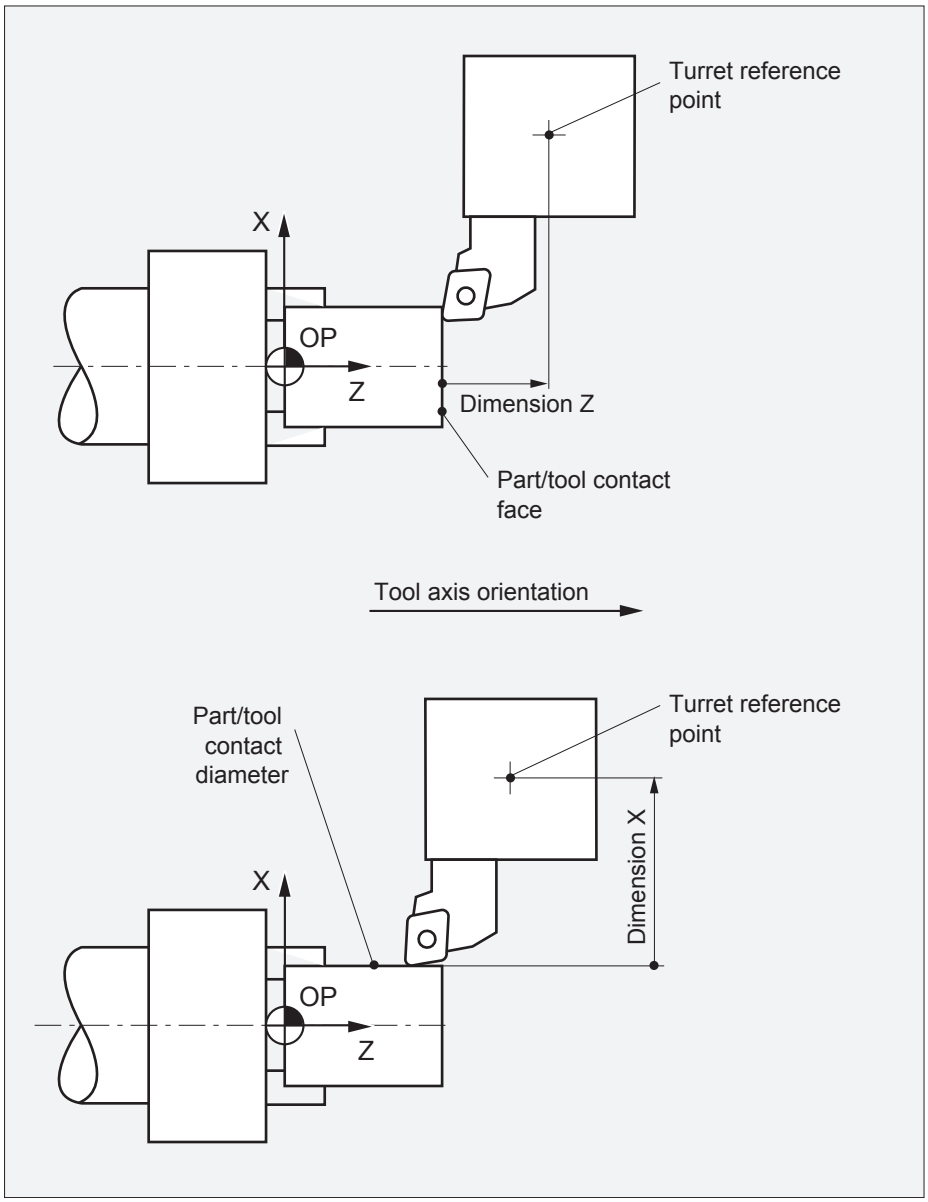


Tool radius = R
Tool length = L

Cutter tip radius = @

2.2.5.2 Turning tools dimensions

Tool dimension = distance from tool cutting edge to turret reference point



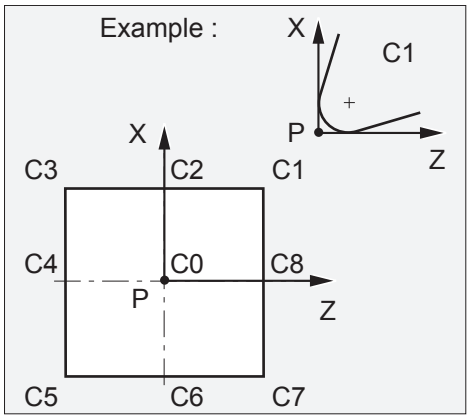
Tool X dimension = X
Tool Z dimension = Z

Definition of Tool Tip Radius and Orientation

The description of a tool is shown below:

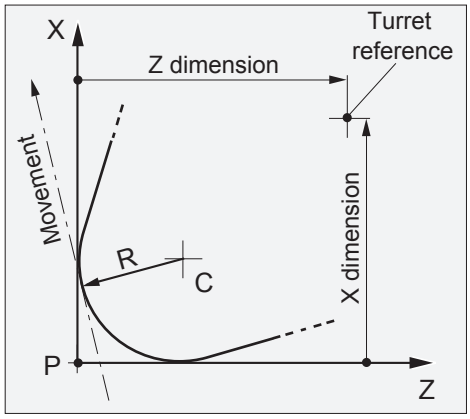
Tool tip orientation = codes C0-C8

The tool tip orientation code allows the system to locate the centre (C) of the tool tip from the theoretical cutting point (P).



Tool tip radius = R

The real cutting point of the tool is obtained by applying a vector of length R perpendicular to the direction of movement from C.



2.2.6 Definition of Dynamic Tool Corrections

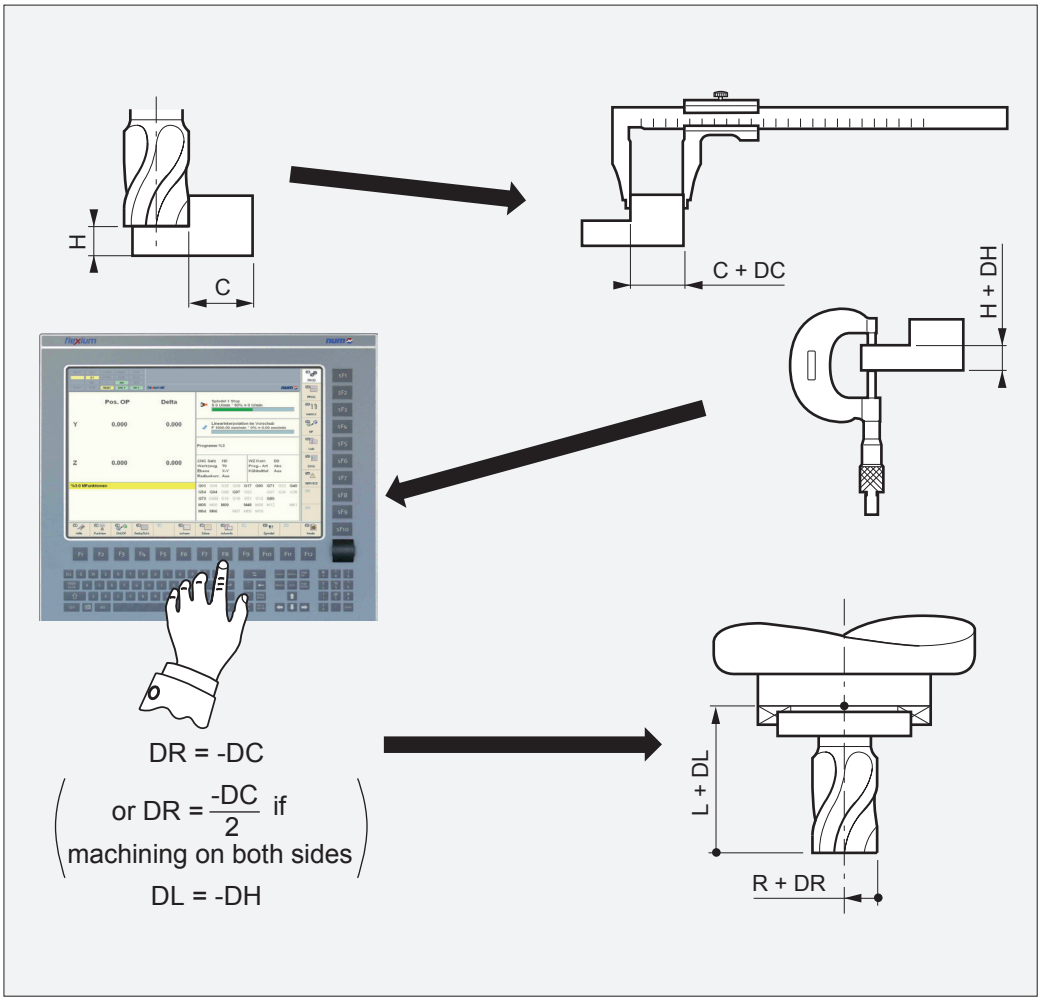
At any time (even during machining), the operator can enter dynamic tool corrections when he observes a difference between the expected and the actual dimensions on a part.

The corrections (positive or negative) compensate for slight variations of the tool or part (wear, expansion). The dynamic tool corrections are taken into account on Length and radius for Milling Machines and on X dimension and Z dimension for Turning Machines.

Dynamic tool corrections are taken into account immediately for a change smaller than 0.1mm. For larger amount they will be taken into account at the end of the prepared blocks.

2.2.6.1 Dynamic Tool Corrections for Milling Tools

Dynamic tool radius correction = DR
Dynamic tool length correction = DL



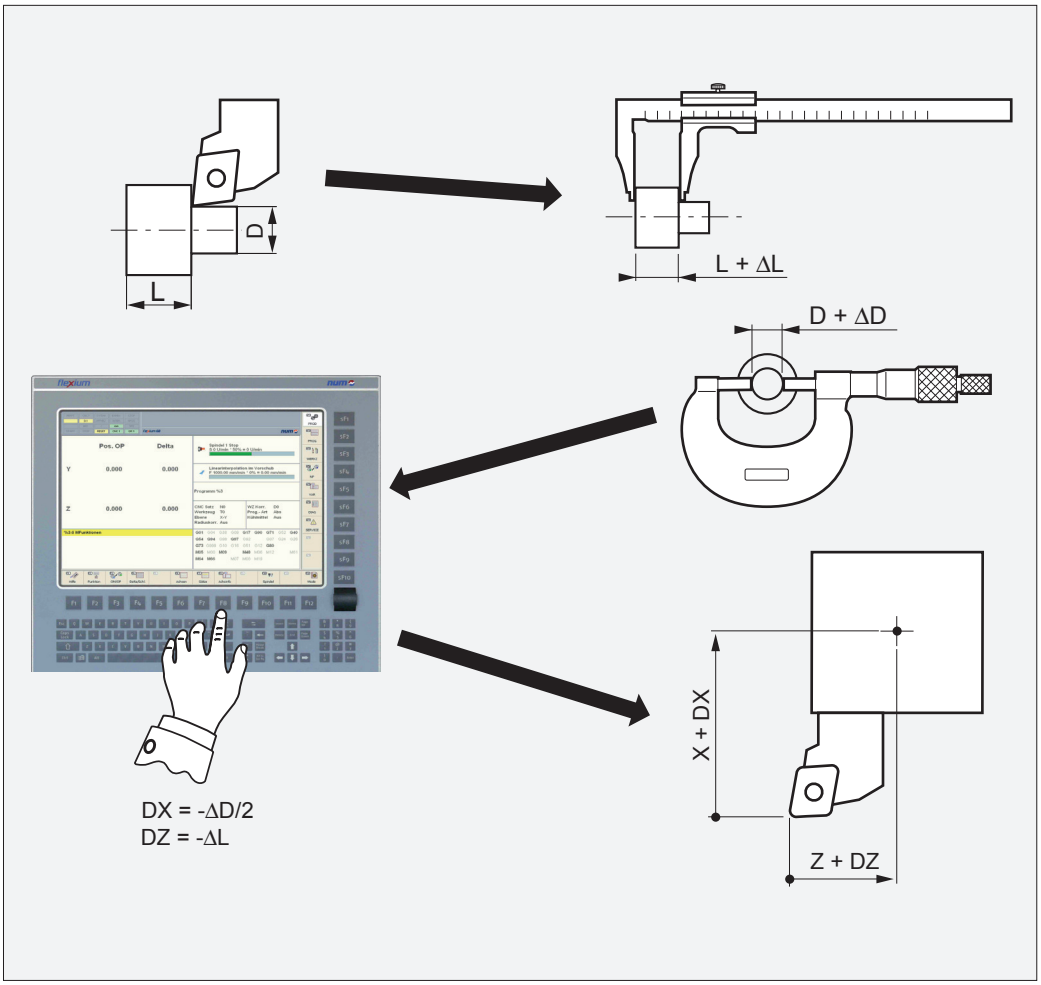
Corrected radius = R + DR
Corrected length = L + DL

2.2.6.2 Dynamic Tool Corrections for Turning Tools

At any time (even during machining), the operator can enter dynamic tool corrections when he observes a difference between the expected and the actual results on a part.

The corrections (positive or negative) compensate for slight dimensional variations of the tool or part (wear, expansion).

Dynamic tool correction on X = DX (diameter)
Dynamic tool correction on Z = DZ



The operator just needs to enter the dimension fluctuations like they are measured (diameter). The system will automatically divide them by 2 to take the right amount at the diameter.

Corrected length on X = X dimension + DX/2
Corrected length on Z = Z dimension + DZ

Page deliberately empty

Summary

3 Structure of a Program 39
3.1 Word Format 40
3.1.1 General Word Format 40
3.1.2 Special Features for Dimensions 40
3.1.3 Internal Unit for Linear Axes 41
3.1.3.1 Correspondence between the word format and internal unit for linear axes 41
3.1.4 Internal Unit for Rotary Axes 41
3.1.4.1 Correspondence between the word format and the internal system unit for rotary axes 41
3.1.5 Examples of word formats 42
3.1.5.1 Word defining a dimension, address X (internal unit in µm) 42
3.1.5.2 Word defining a function, address G 42
3.2 Block Format 43
3.2.1 Examples of blocks 43
3.2.1.1 A block defining a tool change and calling up the tool correction 43
3.2.1.2 A block defining spindle rotation 44
3.2.1.3 A block defining a move 44
3.3 General Structure of a Program 45
3.3.1 General 45
3.3.2 Branches and Subroutine Calls 46
3.3.3 Program Numbering 47
3.3.3.1 Inhibiting display of subroutines being executed 47
3.4 Classification of Preparatory and Miscellaneous Functions 48
3.4.1 Classification of G Functions 48
3.4.1.1 Modal G Functions 48
3.4.1.2 Nonmodal G Functions 48
3.4.1.3 G Functions Incompatible with the state of the program 48
3.4.1.4 G Functions Associated with Arguments 49
3.4.2 Classification of M Functions 50
3.4.2.1 Modal M Functions 50
3.4.2.2 Nonmodal M Functions 50
3.4.2.3 «Pre» M Functions 50
3.4.2.4 «Post» M Functions 50
3.4.2.5 Decoded M Functions 51
3.4.2.6 Encoded M Functions 51

Page deliberately empty

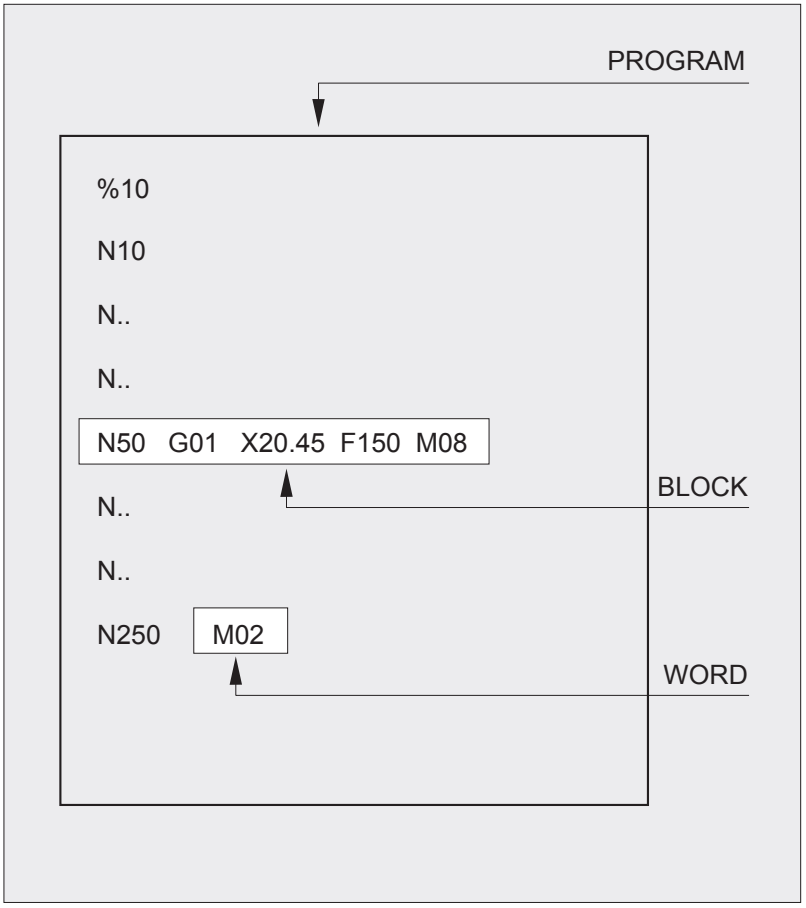
3 Structure of a Program

A CNC part program is a list of instructions and data to be transmitted to the control system.

The creation of a program consisting of blocks and words must obey structure, syntax and format rules.

This structure, syntax and format rules respect the ISO codes and standards:

- ISO (International Standards Organization) 6983-1 (NF Z 68-035), 6983-2 (NF Z 68 036) and 6983-3 (NF Z 68-037).



3.1 Word Format

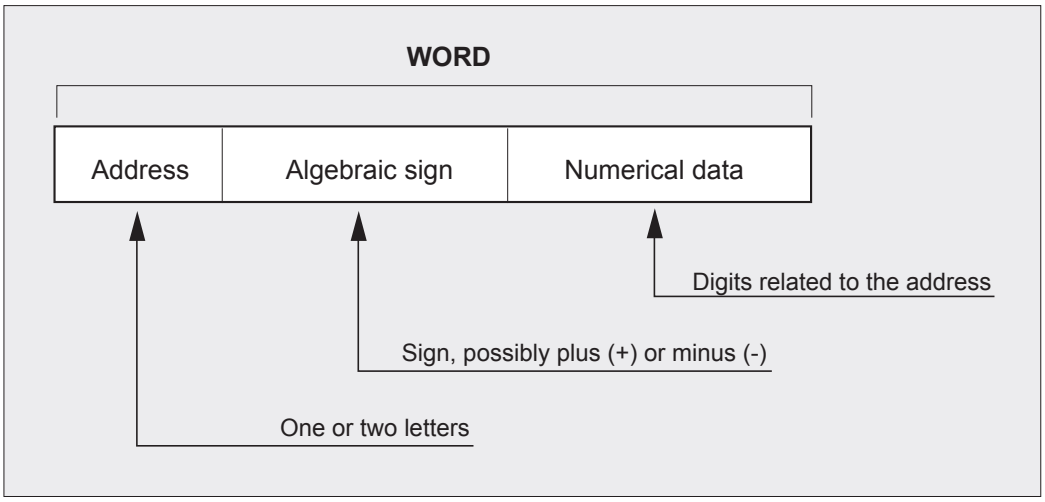
A word contains an instruction or data to be transmitted to the control system.

Word types:

- words defining dimensions
- words defining functions.

The word format defines the specific characteristics of each code word used in programming (see chapter 1).

3.1.1 General Word Format



The number of characters and spaces in a block must not exceed 118.

3.1.2 Special Features for Dimensions

The format of a word used to define a dimension is linked to the internal dimension unit which is specified by the OEM.

Internal dimension units are specified for:

- Linear axes
- Rotary axes.

3.1.3 Internal Unit for Linear Axes

Internal unit is always metric with the mm as a base regardless of the programming unit metric or imperial. The number of decimal digits available for programming linear axes is as follows:

3.1.3.1 Correspondence between the word format and internal unit for linear axes

Minimal programming increment	Word format (metric)	Word format (imperial)
0.1 mm	Format +071	Format +062
0.01 mm	Format +062	Format +053
1 µm	Format +053	Format +044
0.1 µm	Format +044	Format +035
0.01 µm	Format +035	Format +026

Format 0XY means: default sign positive, leading zeros optional, X digits before the decimal point.

3.1.4 Internal Unit for Rotary Axes

The number of decimal digits available for programming rotary axes (base unit is always the degree) is as follows:

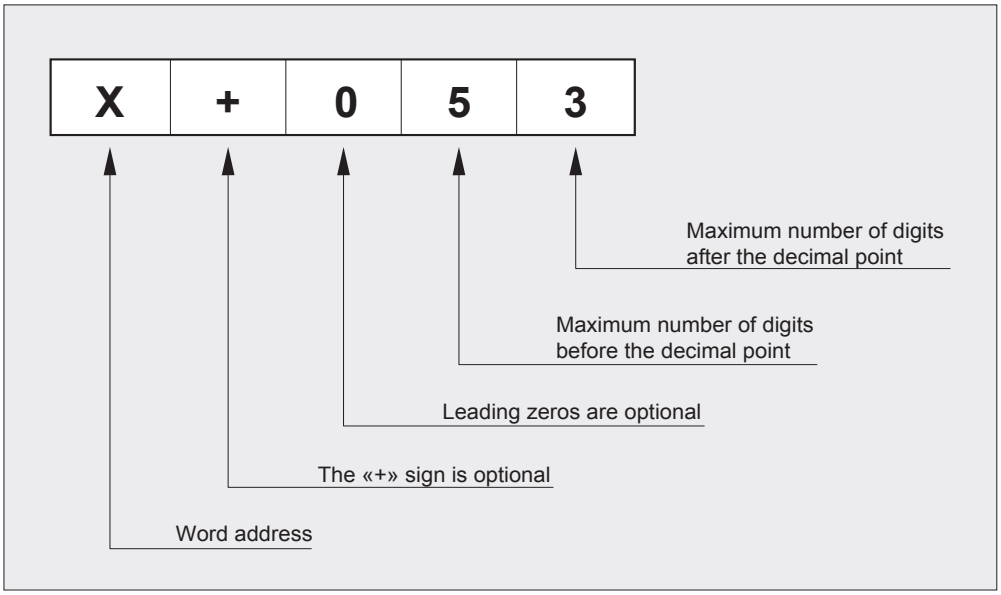
3.1.4.1 Correspondence between the word format and the internal system unit for rotary axes

Minimal programming increment	Definition	Word format
0.1 degree	1 decimal digit	Format 031
0.01 degree	2 decimal digits	Format 032
0.001 degree	3 decimal digits	Format 033
0.0001 degree	4 decimal digits	Format 034

In most cases the internal measuring unit is the same as as the internal programming unit. However the machine builder may have set them at two different values mostly to increase accuracy under specific circumstances and conditions. This might have important consequences in parametric programming and will be detailed in the relevant chapters.

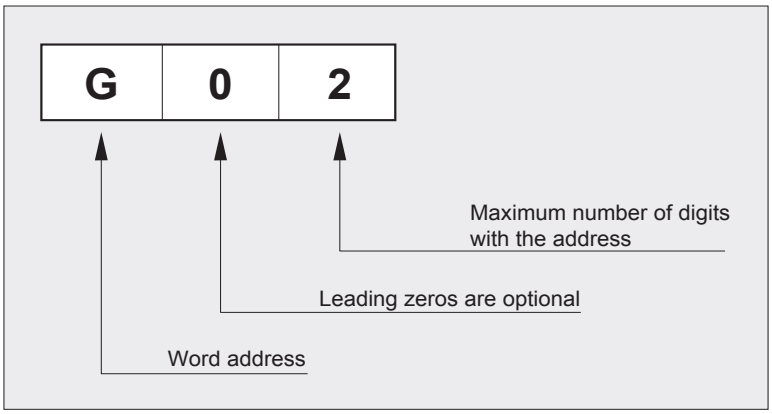
3.1.5 Examples of word formats

3.1.5.1 Word defining a dimension, address X (internal unit in µm)



The dimension 0.450 mm in X+053 format (variable word format), can be written:
X+0.450 or X.45

3.1.5.2 Word defining a function, address G



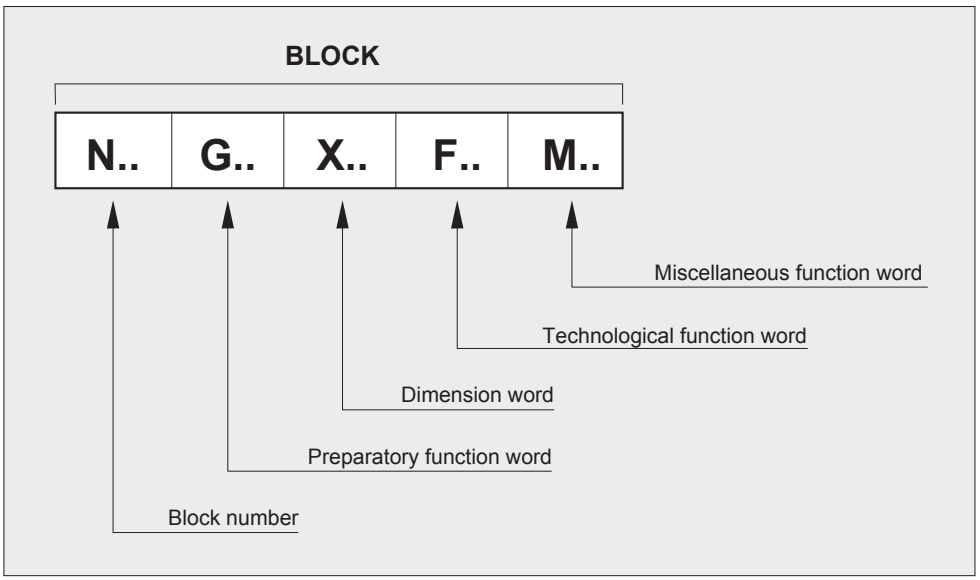
G function words in G02 format (variable word format).

- Word G01 can be written: G1
- Word G04 can be written: G4

3.2 Block Format

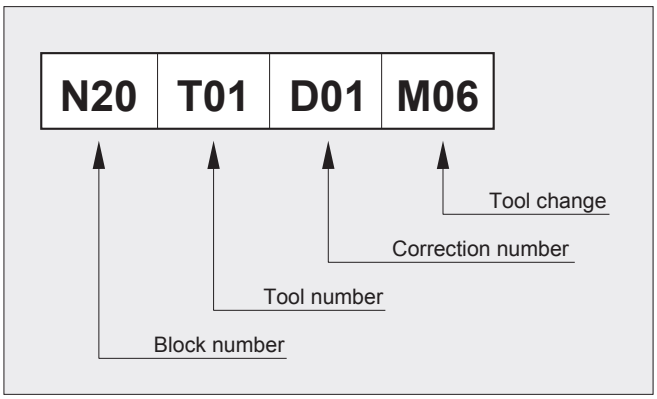
A block (or sequence) is a list of instructions that will be executed by the control system either simultaneously or in a predefined order.

The order of execution and interference between functions will be detailed later in this chapter.

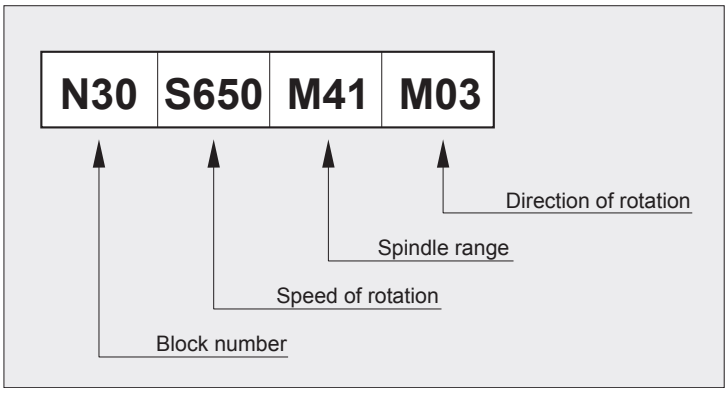


3.2.1 Examples of blocks

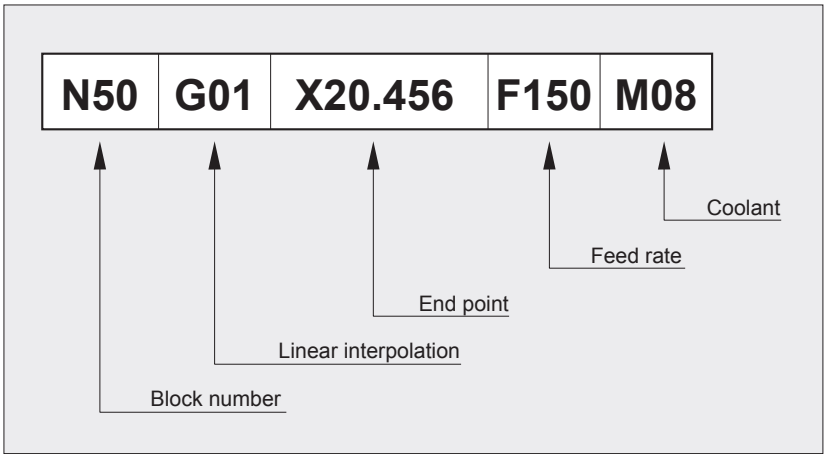
3.2.1.1 A block defining a tool change and calling up the tool correction



3.2.1.2 A block defining spindle rotation



3.2.1.3 A block defining a move



3.3 General Structure of a Program

3.3.1 General

An NC program always start with a "%" character.

A program is generally executed in the order in which the blocks are written. The different blocks can be numbered or not. The order or the sequence of ordering has no effect on the execution with one particular exception explained below.

A block can contain Jump and Branch instructions which will change the execution order. The target for a Jump is always searched from the beginning of the program.

In this case if a Jump is programmed to a block number defined twice, always the first occurrence will be found. There is no error detectd.

A block may contain a comment. An open round bracket "(" signifies the start of that comment. A single close bracket ")" character closes the comment.

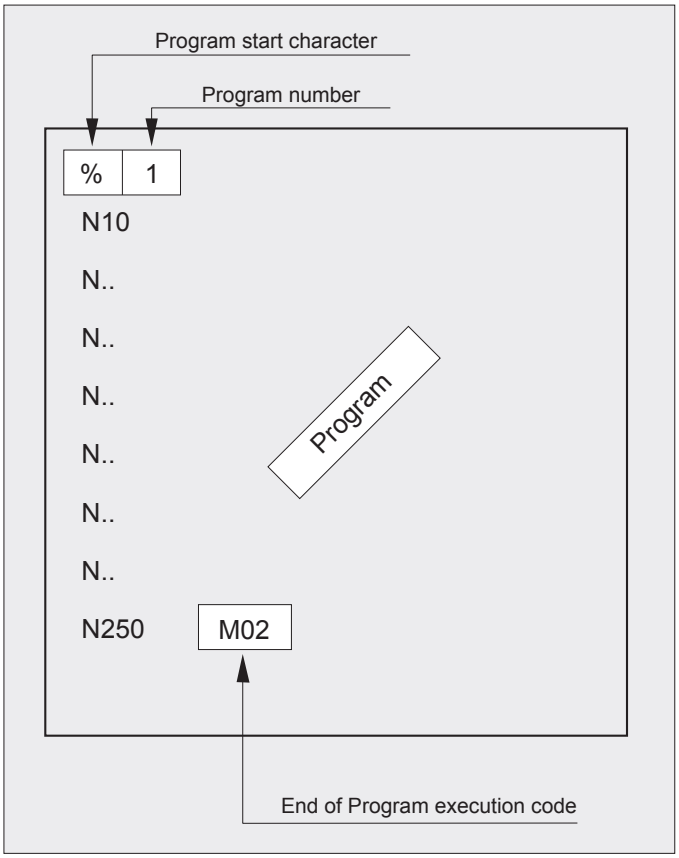
The special character % used to identify the start of a program must not be used in a comment.

Structure of an ISO Program

Program start: % character

Program end: code **M02**

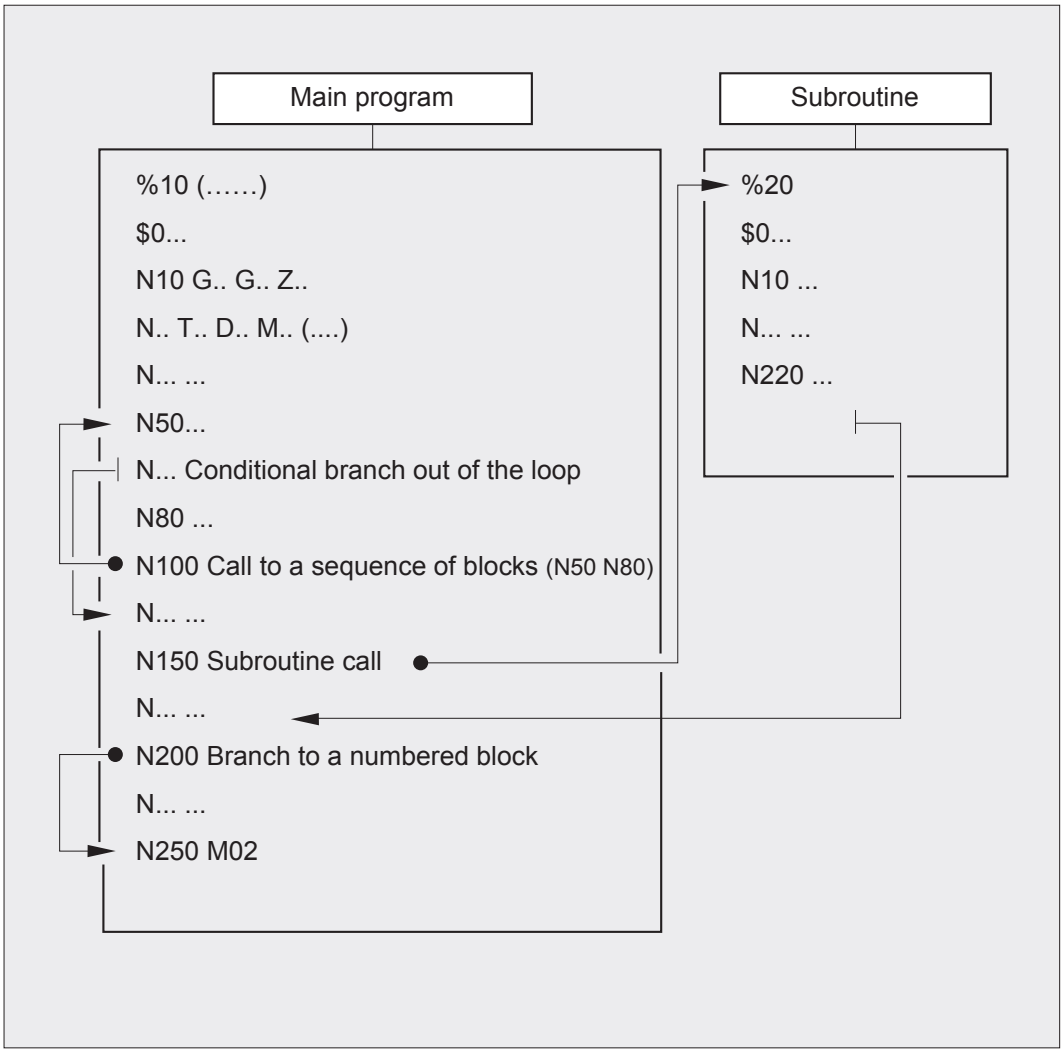
M02 marks the end of program execution, it will generate a RESET. Therefore it should not be used in Sub-Programs (Macros) and is not mandatorily the last sequence of a program as subroutines sequences can be inserted here.



3.3.2 Branches and Subroutine Calls

Particular instructions (branches and subroutine calls) can modify the order in which a program is executed. Branches can be conditional or not.

A program can be structured as follows:



3.3.3 Program Numbering

Program number: The permissible format is %051.

The % character is followed by a program number and possibly by a comment in brackets.

Example

```
%324 (PART No. 72 - PROG 3)
```

A program number can be indexed.
(indices .1 to .8 with multiple channel programming, see chapter 5.24)

Example

```
%425.2 (PROG FOR Channel 2)
```

It is not possible to program ISO function on the same line as the program number.

They will be ignored.

Programs with numbers above %9000 may be used by NUM and the OEM integrating the NC on the machine (check with NUM or the OEM for possible use of these numbers).

The NUM or OEM specific programs are located in a different memory area. The search priority is User area first then OEM area and finally NUM area. Therefore if you create a program number already used by the OEM or NUM, the later will never be executed probably causing some machine malfunction. No error is detected as this is a volontary feature allowing to overwrite for customisation pre-existing cycles.

3.3.3.1 Inhibiting display of subroutines being executed

The content of a subroutine and its internal subroutines can be masked during the execution (E.G. Long calculation sequence which could just fill the screen with difficult to understand data).

To mask the content it is necessary to insert a colon ":" after the program number : e.g. %110: (hidden execution)

The subroutine can still be edited. The data are just masked during execution and replaced by the calling block.

3.4 Classification of Preparatory and Miscellaneous Functions

3.4.1 Classification of G Functions

The G functions, also named preparatory functions are of two types:

- Modal G functions, stay active as long as they are not cancelled by an other G function or a reset
- Nonmodal G functions, active in the current block only.

Moreover it is important to notice that:

- Certain G functions can be incompatible with the current state of the execution.
- Certain G functions must be programmed with the associated arguments in the same block (the order of declaration can be important).

3.4.1.1 Modal G Functions

Functions belonging to a family and that cancel one another (e.g. interpolation).

Certain families of G functions include a default function that is initialised at power up or after a reset. (see chapter 1.6).

These functions remain enabled until cancelled by another function of the same family.

Example for Milling

`N.. G00 X.. Y..`	High-speed linear interpolation
`N.. G01 Z..`	G00 cancelled by linear interpolation at machining feed rate

Example for Turning

`N.. G00 X.. Z..`	High-speed linear interpolation
`N.. G01 Z..`	G00 cancelled by linear interpolation at machining feed rate

3.4.1.2 Nonmodal G Functions

Functions enabled only in the block where they are programmed (cancelled at the end of the block).

Example for Milling and Turning

`N.. G09 X..`	Accurate stop at end of block cancelled at end of block.

3.4.1.3 G Functions Incompatible with the state of the program

Functions whose programming is enabled or not according to the state of the current program.

Example for Milling

`N.. G18 G41 X.. Y..`	Syntax correct, ZX plane selection (G18), followed by cutter compensation (G41)
`N..`	
`N.. G41 G18 X.. Y..`	Syntax incorrect, change of plane prohibited when in cutter compensation.

Example for Turning

`N.. G21 G42 X.. Y.. Z..`	Syntax correct, change of X Y Z coordinate system (G21), followed by radius offset (G42)
`N..`	
`N.. G42 G21 X.. Y.. Z..`	Syntax incorrect, change of coordinate system prohibited with radius offset.

3.4.1.4 G Functions Associated with Arguments

Some function are associated with specific arguments.The argument (s) must immediately follow the function. The analysis of the arguments of a G function is ended by reading a word that does not belong to the list of arguments of this function.

Example for Milling

```
N.. G04 F2 T03 F200
```
Syntax correct

```
N.. G04 T03 F2 F200
```
Syntax incorrect, argument F2 does not immediately follow G04

Example for Turning

```
N.. G04 F2 T03 F0.2
```
Syntax correct

```
N.. G04 T03 F2 F0.2
```
Syntax incorrect, argument F2 does not immediately follow G04.

When a G functions has several arguments, they can be programmed in any order except for G functions that introduce breaks in the sequencing (G10, G76, G77 and G79, see chapter 5.15). The arguments associated with a function can be:

- compulsory,
- optional.

Compulsory Arguments

The arguments are compulsory if:

- the G function serves only to announce arguments.

Example:

```
N.. G16 P+
```
G function and its argument P+

- the G function cancels a former modal state and characterises its argument differently.

Example:

```
N.. G94 F100
N..
N.. G95 F0.5
```
Feed in mm/min

The change from feed in mm/min to mm/revolution requires redefining argument F

Optional Arguments

The arguments are optional if the G function allows them to be defined by default.

Example:

```
N.. G96 [X..] S150
```
Case where the X position (with respect to OP) was specified by an earlier block (G96: constant surface speed for a turning machine)

Arguments Programmed Alone

The argument can be programmed alone in a block when the associated G function is still active.

Example for Milling

```
N.. G94 F150 X.. Y..
N..
N.. X.. Y.. F100
```
Feed in mm/min

Example for Turning

```
N.. G94 F150 X.. Z..
N..
N.. X.. Z.. F100
```
Feed in mm/min

Function G94 is not compulsory with its argument because the system is still in state G94.

3.4.2 Classification of M Functions

The M functions, also named miscellaneous functions are also from two types:
- Modal M functions,
- Nonmodal M functions.

M functions can be:
- «Pre», «Post» or "On the fly" functions,
- decoded or encoded functions (non relevant for part programming).

3.4.2.1 Modal M Functions

Functions belonging to a family of M functions that cancel one another (e.g. Spindle rotation).

Certain families of M functions include a default value function that is initialised at power up or after a reset. (see chapter 1.7).

These functions remain enabled until they are cancelled by another function of the same family.

Example for Milling and Turning:

`N.. S500 M03`	Start of spindle rotation
`N.. M05`	Spindle stop, cancels M03

3.4.2.2 Nonmodal M Functions

Enabled only in the block where they are programmed.

Example for Milling and Turning:

`N.. M00`	Program stop

3.4.2.3 «Pre» M Functions

Functions executed before axis movements programmed in the block.

Example for Milling:

`N.. X100 Z50 M08`	Coolant function M08 is executed before the movements on X and Y.

Example for Turning:

`N.. X100 Y50 M08`	Coolant function M08 is executed before the movements on X and Z.

3.4.2.4 «Post» M Functions

Functions executed after the axis movements programmed in the block.

Example for Milling:

`N.. X50 Y100 M09`	The coolant off function (M09) is executed after movements on X and Y

Example for Turning:

`N.. X100 Z100 M08`	The coolant off function (M09) is executed after movements on X and Z.

3.4.2.5 Decoded M Functions

The decoded M functions are the basic system functions whose meaning and behavior are known.

Each one can be either Pre or Post, Modal or non Modal.

Generally Activation function are "Pre" where deactivation is "Post". For details, see chapter 1.

Example for Milling and Turning:

```
N.. T01 M06                    Tool change function M06
```

Several decoded M functions can be programmed in the same block.

Example for Milling and Turning:

```
N.. G97 S500 M03 M40 M08
```

3.4.2.6 Encoded M Functions

The encoded functions are defined by the machine manufacturer and are specific to the machine (See manufacturer's technical data).

The modality as well as the exclusion is managed by the machine manufacturer. They include all M codes except those already defined as Decoded.

They have two functionalities according to the function number.

Encoded Functions M100 to M199

These functions are "Post" functions, they require an acknowledgement (which can be automatic).

Only one such function can be programmed in a block. If more than one function is programmed, only the last one in the block will be taken into account.

Encoded Functions M200 to M899

These functions are modal «pre» functions. The program continues without waiting for the execution report. These functions are also called "On the Fly" M functions as they are executed at the beginning of the block without interruption of the movement.

They can not require an acknowledgement.

Only one of these function can be programmed in a block. If more than one is programmed only the last one will be taken into account.

They can however be programmed on the same block that other types of M function (decoded and/ or encoded up to 199).

Only one of these functions is allowed in a part program block.

As these function do not require acknowledgement use caution when programming several M "On the fly" consecutively. Be sure that there is enough time between each (dwell or axes movement) for the machine to take them into account. Refer to the machine operation manual.

These are the standard ranges (100 to 199 and 200 to 899). The machine builder has the possibility to modify the frontier (e.g. 100 to 499 and 500 to 899). The limit of M899 being constant.

Page deliberately empty

Summary

4 Axis Programming 55
4.1 General 55
4.2 Programming the Independent Secondary Axes 56
4.3 Programming of Carrier/Carried Parallel Axis Pairs 56
4.4 Programming Rotary Axes Modulo 360 Degrees 57
4.4.1 Rotary axis programmed in absolute dimensions (G90) 57
4.4.2 Rotary axis programmed in incremental dimensions (G91) 58
4.5 Programming Rotary Axes with Limited Travel 59
4.6 Programming of Axes A, B or C Declared as Nonrotary 59

Page deliberately empty

4 Axis Programming

4.1 General

Programmable axes:
- Primary axes X, Y, Z,
- Secondary axes U, V, W,
- Rotary axes A, B, C.

Primary and secondary axes:
- they can be independent or form carrier/carried axes pairs,
- they can be programmed in millimetres (basic unit) or inches.

Rotary axes:
- They can be modulo 360 degrees or have limited travel,
- They are programmed in degrees (basic unit).

Reminder

Definition of the Internal System Measurement Units

The internal measurement unit is defined by the OEM when integrating the CNC. It directly affects the machine travels, the minimal positioning step and the speeds.

The number of decimal digits is also defined by the OEM, it can be different from the internal measurement and determines the word formats (see chapter 1.10 and chapter 3.1).

For linear axes, the internal unit can be 0.1 mm, 0.01 mm, µm, 0.1 µm or 0.01 µm.

For rotary axes, the internal unit can be 0.1 degree, 0.01 degree, 0.001 degree or 0.0001 degree.

For ISO functions and programming arguments defining angular values (EA.., EC.., ED.., etc.), the unit is always 0.0001 degree.

For additional information, refer to the machine operation manual.

4.2 Programming the Independent Secondary Axes

Programming the independent secondary axes U, V, W is unrelated to the programming of the primary axes X, Y, Z.

The secondary axes U, V, W are respectively parallel to the primary axes X, Y, Z.

It should be noted that the tool length correction (L for Milling Machine offsets and X, Z for Turning Machine offsets) is not applied on the secondary axes.

The tool radius correction is applied.

4.3 Programming of Carrier/Carried Parallel Axis Pairs

Axes pairs (X/U,Y/V, Z/W) can be declared by the OEM as Carrier/Carried.

The tool tip position reflects the cumulated position of both axes. Only one axis of each pair can be programmed in a single block

The tool length compensation is taken on the primary axis of the pair (X, Y, Z).

4.4 Programming Rotary Axes Modulo 360 Degrees

4.4.1 Rotary axis programmed in absolute dimensions (G90)

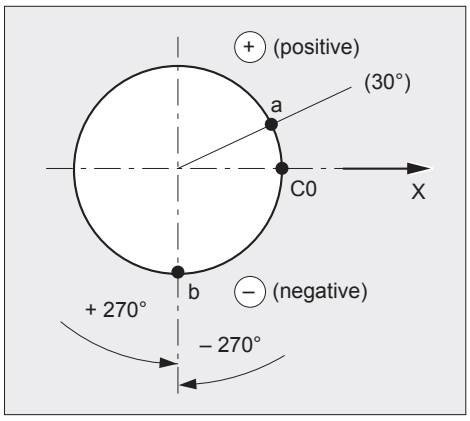
The programmed value assigned to the axis is the end angular position with respect to the program origin, regardless of the sign (value between 0 and 360 degrees, maximum one revolution) .

The sign (+ or -) determines the direction of rotation used to reach this point.

Example

a : Start point.

b : End point.



Positive movement

```
N.. ...
N.. G90 B+270
N..
```

Negative movement

```
N..
N.. G90 B-270
N..
```

Both sequence of blocks will reach the same angular position.

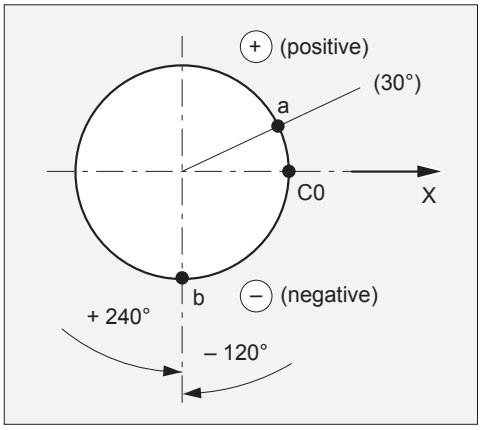
4.4.2 Rotary axis programmed in incremental dimensions (G91)

The programmed value assigned to the axis indicates the amplitude of rotation with respect to the previous position (see chapter 5.3 for function G91).

Example

a : Start point.

b : End point.



Positive rotation

```
N.. ...
N.. G91 B+240
N..
```

Negative rotation

```
N.. ...
N.. G91 B-120
N..
```

With incremental programming G91, a movement of more than one revolution is allowed on modulo rotary axes A, B or C.

The maximum amplitude is 5759.9999 degrees (almost 16 revolutions). If this value is exceeded, the system returns error message 1.

Caution: remember that the first movement in a program should always be absolute (G90) to correctly set the position. If the first movement is in incremental, it will be considered as absolute anyway.

In Turning Machine configuration only (G20) please note that the speed on the C axis is related to the position of the X axis. The programmed speed will be applied in mm/min at the tool contact point.

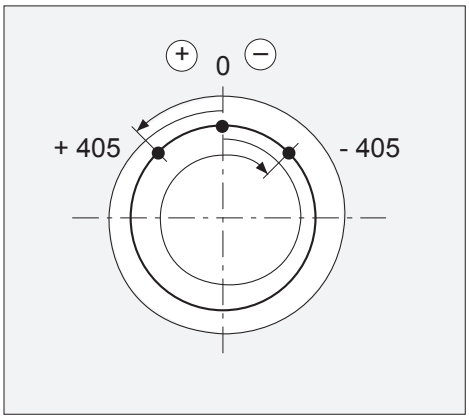
4.5 Programming Rotary Axes with Limited Travel

Rotary axes A, B or C with limited travel are defined by the OEM like linear axes and therefore follow the same programming rules, in particular the sign defines the final coordinate and not the direction of travel.

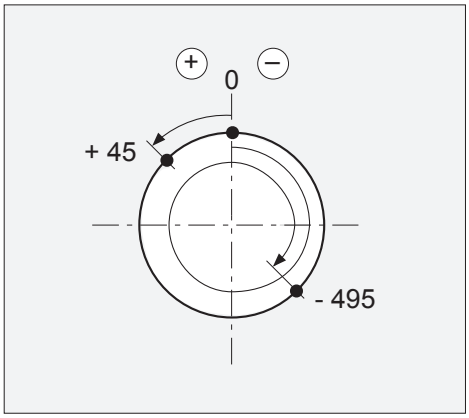
A Travel for such an axis can be from a fraction of revolution to several revolutions. The maximum limit being +/- 9999.9999 degrees. For the current limits see the machine operation manuals.

Example

Rotation by more than one revolution in absolute dimensions (G90).



Rotation by more than one revolution in incremental dimensions (G91).



4.6 Programming of Axes A, B or C Declared as Nonrotary

When axes A, B or C are declared as nonrotary, they are considered as linear axes (in particular in keyboard Homing mode and Shift mode).

The speed of movement on axes A, B or C declared as nonrotary is expressed in mm/min.

However, if they are programmed in a block together with primary and secondary axes X, Y, Z, U, V or W, the programmed speed is assigned to the latter.

Page deliberately empty

Summary

5 ISO Programming 67
5.1 G Functions, M Functions and other Functions - Main summary tables 67
5.1.1 G Functions - Main summary table 67
5.1.2 M Functions - Main summary table 79
5.1.3 Other Functions - Main summary table 83
5.1.4 Word Format - Main summary table 86
5.2 Programming with Reference to Diameter or Radius 88
5.2.1 System Programmed with Reference to Diameter 88
5.2.2 System Programmed with Reference to Radius 88
5.3 Absolute or Incremental selection ► G90, G91 89
5.4 Plane Selection ► G17, G18, G19 91
5.5 Spindle Control 92
5.5.1 Spindle Direction of Rotation ► M03, M04, M05 92
5.5.2 Spindle Speed Control ► G97 94
5.5.2.1 Review of surface speed 96
5.5.2.2 Using a fixed spindle rotation speed (N) 96
5.5.2.3 Using a constant surface speed (V) 97
5.5.2.4 Spindle speed limit 97
5.5.3 Constant Surface Speed on Turning machine ► G96 98
5.5.4 Spindle Speed Limiting ► G92 99
5.5.5 Spindle Range ► M40, M41, M42, M43, M44, M45 101
5.5.6 Indexed Spindle Stop ► M19 102
5.5.7 Spindle Control Selection ► M64, M65, M62, M63, M61 103
5.5.8 Spindle Measurement Selection ► M66, M67, M68, M69 105
5.6 Rapid Positioning ► G00 106
5.6.1 Positioning with respect to the cutter dia. (R-/R+) example 108
5.7 Programming Movements 110
5.7.1 Linear Interpolation ► G01 110
5.7.1.1 Programming of additional axes and carrier/carried axis pairs 111
5.7.2 Positioning with respect to the cutter dia. (R+/R-) 113
5.7.3 Circular Interpolation ► G02, G03 114
5.7.3.1 Direction of movement according to the turret position with Turning Machine 116
5.7.3.2 Programming a circle by its radius (R..) 117
5.7.3.3 Programming a circle by the coordinates of its centre (I and J, plane G17) 117
5.7.3.4 Programming the additional axes and carrier/carried axis pairs 118
5.7.4 Helical Interpolation 119
5.7.5 Determination of the helix depth 120
5.7.5.1 Syntax according to the plane selected: G17/G18/G19 121
5.7.5.2 Programming of additional axes and carrier/carried axis pairs 121
5.7.6 Circular Interpolation Defined by Three Points ► G23 123
5.7.6.1 Syntax According to the Plane Selected: G17/G18/G19 124
5.7.7 Polar Programming 125
5.7.7.1 Polar Programming of a Line 126
5.7.7.2 Polar Programming of a Circle 128
5.7.7.3 Defining a Circle by the Arc Angle 132
5.7.8 Programming Fillets and Chamfers 138
5.7.8.1 Fillet between two interpolations ► EB+ 138
5.7.8.2 Chamfer Between Two Linear Interpolations ► EB- 139
5.7.9 Accurate End of Block Stop ► G09 141
5.8 «Inclined Axis» or «Inclined Wheel» 143
5.9 Initial Tool Positioning Before Machining on an Inclined Axis ► G07 145
5.10 Movement Along the Inclined Axis ► G05 146
5.10.1 Feed Rate 148

5.10.1.1 Feed Rate Expressed in Millimetres, Inches or Degrees per Minute ► G94 148
5.10.1.2 Programming additional axes 149
5.10.1.3 Programming of a modulo rotary axis programmed alone 149
5.10.1.4 Programming of modulo rotary axis and a linear axis 149
5.10.1.5 Programming of two modulo rotary axes 149
5.10.1.6 C axis feed rate for Turning Machine 150
5.10.2 Inverse Time Feed Rate Coding (V/D) ► G93 151
5.10.3 Feed Rate Expressed in Millimetres or Inches per Revolution ► G95 155
5.10.4 Tangential Feed Rate ► G92 R.. 157
5.10.5 Setting feed rate per axis set ► G92 F 160
5.10.6 Feed Rate Specific to Fillets EB+ and Chamfers EB- ► EF 161
5.11 Tool Programming 164
5.11.1 Tool Change ► M06 164
5.11.2 Tool Axis Orientation ► G16 167
5.11.3 Tool Offset activation ► D 170
5.11.3.1 Tool length correction (L) 171
5.11.3.2 Tool dimensions corrections 172
5.11.3.3 Tool radius corrections 172
5.11.3.4 Tool tip orientation (C) according to the turret position 174
5.11.3.5 Use of tool corrections (X, Z, R, C) on a lathe with two mechanically coupled turrets . 174
5.11.4 Tool radius compensation ► G41, G42, G40 176
5.11.4.1 Tool positioning 179
5.11.4.2 Radius correction according to the turret position 180
5.11.4.3 Tool outside the profile (line/line or circle/circle) 181
5.11.4.4 Tool outside the profile (line/circle) 183
5.11.4.5 Tool inside the profile 183
5.11.4.6 Tool inside the profile, special cases 184
5.11.5 Tool disengagement 184
5.11.6 Tridimentional tool correction with toroid or spherical tool ► G29 195
5.11.6.1 Concept of Surface and Normal Vector 197
5.11.6.2 Toroidal and Spherical Tools 198
5.11.6.3 Double Twist Machine Head 198
5.11.6.4 Geometric Transformations with 3-Axis Correction 199
5.11.6.5 Geometric Transformations with 5-Axis Correction 201
5.11.6.6 Tridimentional Tool Correction with Cylindrical Tool ► G43 202
5.12 Canned Cycles 203
5.12.1 Cycle Overview ► G8x 203
5.12.2 Cancellation of a Canned Cycle ► G80 205
5.12.3 Centre Drilling Cycle ► G81 206
5.12.4 Counterboring Cycle ► G82 208
5.12.5 Peck Drilling Cycle ► G83 210
5.12.6 Tapping Cycles ► G84 214
5.12.7 Rigid Tapping Cycle ► G84 217
5.12.8 Rigid Tap Removal ► G39+ 221
5.12.8.1 Execution of the Rigid Tap Removal Cycle 222
5.12.9 Reaming Cycle ► G85 223
5.12.10 Boring Cycle with Indexed Spindle Stop at the Bottom of the Hole ► G86 225
5.12.11 Drilling Cycle with Chip Breaking ► G87 227
5.12.12 Boring and Facing Cycle ► G88 231
5.12.13 Boring Cycle with Dwell at the Bottom of the Hole ► G89 233
5.12.14 Thread Chasing Cycle ► G31 238
5.12.14.1 Cycle repeat to achieve the thread precision 240
5.12.15 Cycle summary 241
5.12.16 Examples of G81 to G89 cycles 242
5.13 Turning specific canned cycles 245
5.13.1 Constant Pitch Thread Chasing Cycle ► G33 245
5.13.1.1 Tool retraction slope at the end the pass defined by R 247
5.13.1.2 Penetration defined by EB 248
5.13.1.3 Tapered thread cutting, angle defined by EA 248
5.13.1.4 Multiple start thread cutting 249
5.13.2 Sequenced Thread Cutting ► G38 251
5.13.3 Resume Thread Cutting ► G38+ 253

5.13.3.1 Inclusion of function G38+ in Cycle G33 254
5.13.3.2 Associating a Teach-in Cycle with Resume Thread Cutting Cycle G33 255
5.13.4 Rough Turning/Facing Cycle ► G64 256
5.13.4.1 Roughing cycle P or R 259
5.13.5 Groove Roughing Cycle ► G65 267
5.13.6 Plunging Cycle ► G66 272
5.13.7 Roughing Cycle with Groove ► G63 276
5.13.7.1 Definition of EB and EC 280
5.13.7.2 Cycle differences between turning and facing 281
5.14 Simple Pocket Cycle ► G45 289
5.14.1 Pass setting possibilities 291
5.14.2 Roughing cycle with a single tool 292
5.14.3 Roughing Cycle Steps 292
5.14.4 Roughing and finishing cycle with a single tool 293
5.14.5 Finishing cycle alone with a new tool 293
5.14.6 Pass setting in the plane (Q) 294
5.14.6.1 Pass setting in the tool axis (P) 294
5.15 Breaks in Sequence ► G77 297
5.15.1 Unconditional Branch to a Subroutine or Sequence of Blocks with Return 297
5.15.2 Subroutine Branch by M Function ► M 303
5.15.2.1 Specific Feature of a Call to a Subroutine Number Between %11001 to %11999 304
5.15.3 Branch to a Sequence without Return ► G79 305
5.15.4 Subroutine Call by PLC 307
5.15.5 Block Interrupt ► G10 308
5.15.5.1 Analysis of the interrupt 309
5.15.5.2 Special Use of Sequence Interrupt 311
5.15.6 Temporary Suspension of Next Block Preparation ► G79+/- 313
5.15.7 Emergency Retract ► G75 315
5.15.8 Branch to Automatic Homing Subroutine %9990 318
5.15.9 Subroutine Branch on a Reset %11000 319
5.15.10 Restrictions Related to Drip Feed Mode 320
5.15.11 Call to Subroutine Return Block ► G77 -i 321
5.15.12 ISO Program or Block Creation/Deletion ► G76+/- 322
5.15.13 Creating a Program ► G76+ H 323
5.15.14 Deleting a Program ► G76- H 324
5.15.15 Inserting a Block ► G76+ 325
5.15.16 Deleting a Block ► G76- 326
5.16 Movement Referential selection 327
5.16.1 Programming of Movements in Absolute Coordinates Referenced to the Measurement Origin ► G52 327
5.16.2 Datum Shift DAT1 and DAT2 Cancel/Enable ► G53, G54 330
5.16.3 Program Origin Preset ► G92 331
5.16.4 Program Origin Offset ► G59 333
5.16.4.1 Function G59 programmed in absolute dimensions (G90) 334
5.16.4.2 Function G59 programmed in incremental dimensions (G91) 334
5.16.4.3 Features related to arguments I, J (K) 334
5.16.5 Angular offset ► ED.. 342
5.16.6 Table Offset by DAT3 with Milling Machine ► DAT3 346
5.17 Spline Curve Interpolation ► G48, G06, G49 348
5.17.1 Spline Curve Interpolation ► G48 349
5.17.2 Spline Curve Execution Command ► G06 351
5.17.3 Freeing Memory by Deleting a Spline Curve ► G49 356
5.18 Coordinates Systems with C Axis with Turning Machine 357
5.18.1 Programming in Polar Coordinates ► G20 357
5.18.2 Definition of the Start X for Interpolation on the C Axis ► G98 358
5.18.3 Programming in Cartesian Coordinates ► G21 359
5.18.4 Programming in Cylindrical Coordinates ► G22 364
5.19 Other Functions 367
5.19.1 Dwell ► G04 367
5.19.2 Programmed Feed Stop ► M12 368
5.19.3 Feed Increase ► G12 369
5.19.4 Programming in Inches or Metric Data ► G70, G71 370

5.19.5 Axis Clamping and Unclamping ► M10, M11 372
5.19.6 Coolant ► M08, M07, M09 373
5.19.7 Program Stop ► M00 374
5.19.8 Optional Stop ► M01 375
5.19.9 Inhibition of MDI Modes ► M999, M998 376
5.19.10 Forced Block consecutive execution ► M997 378
5.19.11 Potentiometer Inhibit ► M49, M48 379
5.19.12 Block Skip ► / N 380
5.19.13 Acceleration Reduction ► EG 381
5.19.14 Scaling Factor ► G74, G73 382
5.19.15 Mirror function ► G51 386
5.19.15.1 Influence of the mirror function on the A, B or C rotary axes 387
5.19.15.2 Influence of the mirror function on circular interpolations 387
5.19.16 Processing of Blocks and Programmed G and M Functions ► G999, G998, G997 389
5.19.17 Tridimentional curve smoothing ► G104 391
5.20 Message Transmission ► $0 393
5.20.1 Message Transmission to the Display 393
5.21 Message Display with request for an Operator Response ► L../E..,$ 395
5.22 Display of Messages with Parametric Value ► $ 397
5.23 Transfer of the Current Values of L Variables and E Parameters into the Part Program ► G76 398
5.24 Special Programming for Multiple Channels 400
5.24.1 Program Declaration 400
5.24.2 Programming Notes 400
5.24.3 Programming the Spindles 403
5.24.4 Releasing the Current Spindle in the Channel ► M61 404
5.24.5 Channel synchronisation with markers ► G78 405
5.24.5.1 Programming a Meeting Point 407
5.24.5.2 Programming by function G78 407
5.24.5.3 Programming with M functions 408
5.25 Special Programming of Auxiliary channels 409
5.25.1 Program Declaration and Storage 409
5.25.2 Programming of the Auxiliary channels 410
5.25.2.1 Emergency Retraction on an Auxiliary Channel 410
5.25.2.2 Exchanging Axes between Channels 411
5.25.2.3 Exchanging Spindles Between Channels 411
5.26 Special Features of Mixed Machines (MX) 412
5.26.1 Features Related to the Machine Axes 412
5.26.1.1 Boring Machine 413
5.26.1.2 Mixed Type Machine 413
5.26.2 Assignment of Tool Dimensions to the Axes 414
5.26.3 Preselection of the Program Origin (with function G92) 416
5.26.4 Notes on Programming the ISO Functions 417

Page deliberately empty

Page deliberately empty

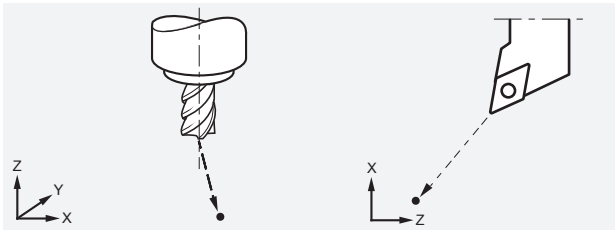
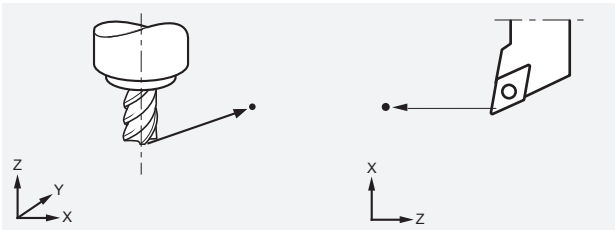
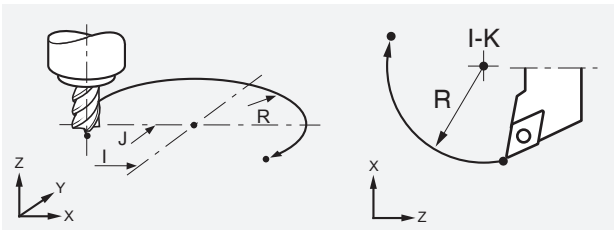
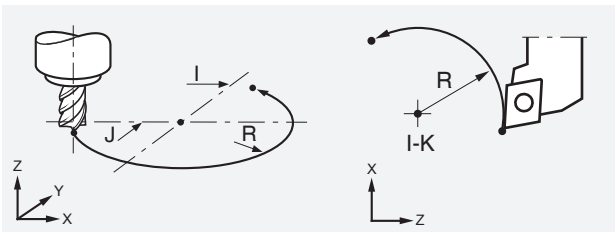
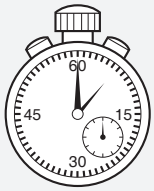
5 ISO Programming

5.1 G Functions, M Functions and other Functions - Main summary tables

5.1.1 G Functions - Main summary table

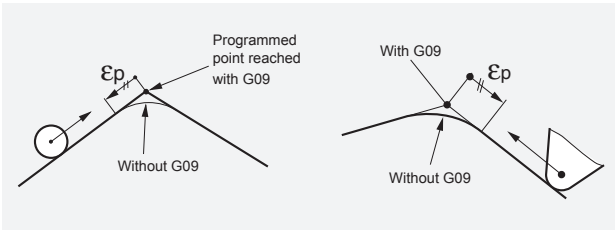
The functions initialized at power on are identified by «*».
Click on **Gxx** to give the detailed information.

The "Syntax" indications in the table are given for Milling application only.
Please refer to the detailed Gxx information to give the "Syntax" on Turning application.

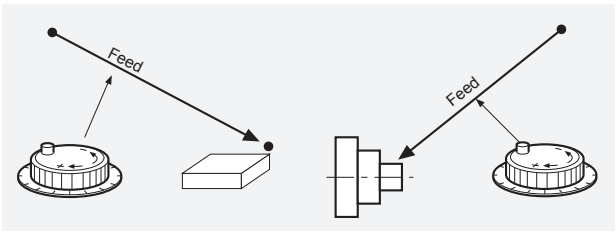
G00	**Linear interpolation at high speed** **Syntax:** N.. [G90/G91] **G00** [R±] X.. Y.. Z.. **Cancellation:** G01/G02/G03	
G01	***Linear interpolation at programmed feed rate** **Syntax:** N.. [G90/G91] **G01** [R±] X.. Y.. Z.. [F..] **Cancellation:** G00/G02/G03	
G02	**Clockwise circular interpolation at programmed feed rate** **Syntax (XY plane):** N.. [G17] [G90/G91] **G02** X.. Y.. I..J.. or R.. [F..] **Cancellation:** G00/G01/G03	
G03	**Counterclockwise circular interpolation at programmed feed rate** **Syntax (XY plane):** N.. [G17] [G90/G91] **G03** X.. Y.. I..J.. or R.. [F..] **Cancellation:** G00/G01/G02	
G04	**G04: Programmable dwell** **Syntax:** N.. **G04** F.. **Cancellation:** End of block	
G05	**Programming a movement on the inclined axis** **Syntax:** N.. [G90/G91] [G00/G01] **G05** X.. **Cancellation:** End of block	

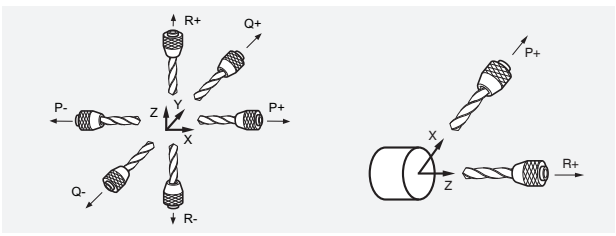
G06	**Spline curve execution command**
	Syntax: N.. **G06** NC..
	Cancellation End of block

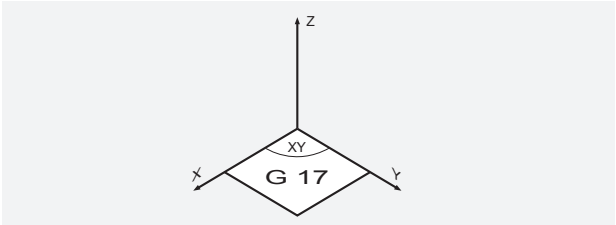
G07	**Initial tool positioning before machining along an inclined axis**
	Syntax: N.. [G90] [G00/G01] **G07** X.. Z..
	Cancellation End of block

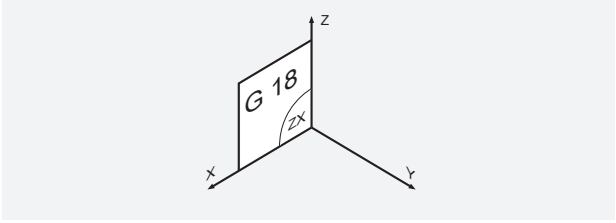
G09	**Deceleration at end of block before continuation on next block**	
	Syntax: N.. **G09** [G00/G01/G02/G03] X.. Y.. Z.. [F..]	
	Cancellation End of block	

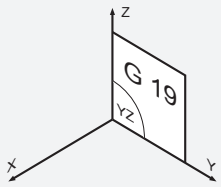
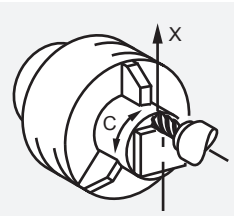
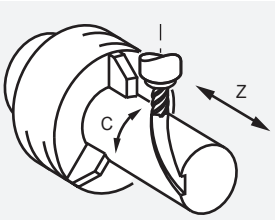
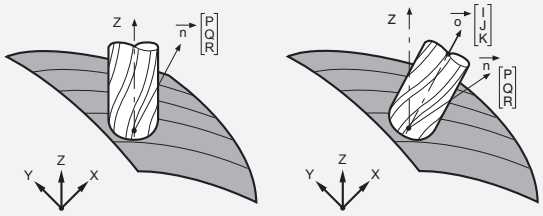
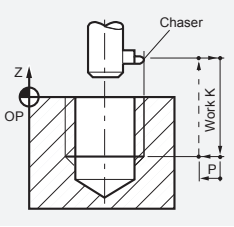
G10	**Interruptible block**
	Syntax: N.. [G40] [G04] [G00/G01/G02/G03] X.. Y.. Z.. **G10** [:n] [+X.. or F..] [@n < > Value] N.. [+ Number] [EF..]
	Cancellation: End of block

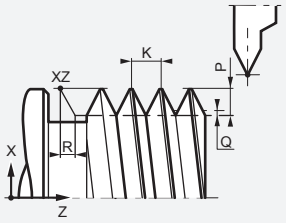
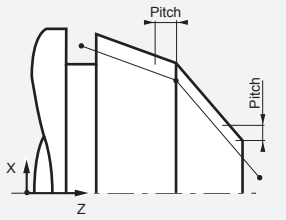
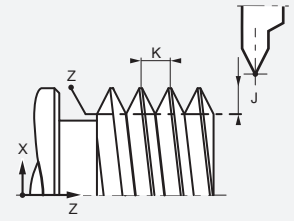
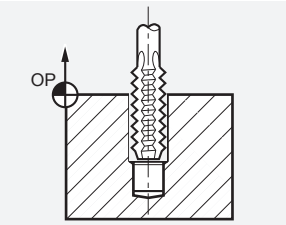
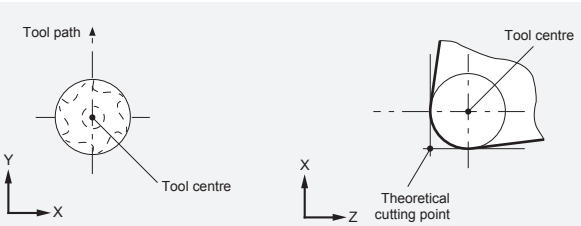
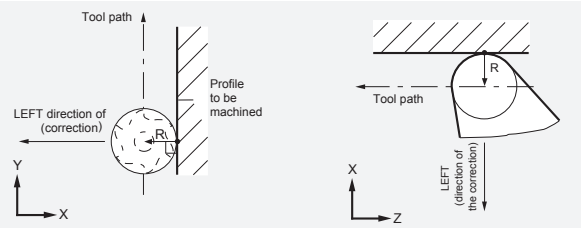
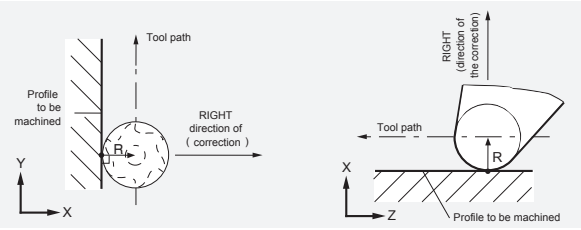
G12	**Overspeed by handwheel**	
	Syntax: N.. [G01/G02/G03] **G12** X.. Y.. Z.. [F..] [$0 ...]	
	Cancellation: End of block	

G16	***Definition of the tool axis orientation with addresses P, Q, R**	
	Syntax: N.. **G16** P±/Q±/R±	
	Cancellation: G16 P±/Q±/R±.	

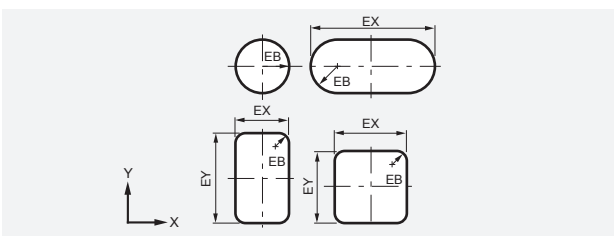
G17	***XY plane selection**	
	Syntax: N.. **G17**	
	Cancellation: G18/G19	

G18	**ZX plane selection**	
	Syntax: N.. **G18**	
	Cancellation: G17/G19	

G19	**YZ plane selection** **Syntax:** N.. **G19** **Cancellation:** G17/G18	
G20	***Programming in polar coordinates (X, Z, C)** **Syntax:** N.. [G40] **G20** [G00/G01] [X.. Z.. C..] [F..] **Cancellation:** G21 and G22	
G21	**Programming in cartesian coordinates (X, Y, Z)** **Syntax:** N.. [G40] **G21** [G00/G01] [G41/G42] [X.. Y.. Z..] [G94 F..] **Cancellation:** **G20 and G22**	
G22	**Programming in cylindrical coordinates (X, Y, Z)** **Syntax:** N.. [G40] **G22** [G00/G01] [G41/G42] [X.. Y.. Z..] [G94 F..] **Cancellation:** **G20 and G21**	
G23	**Circular interpolation defined by three points** **Syntax:** (XY plane) N.. [G17] [G90/G91] **G23** X.. Y.. I.. J.. [F..] **Cancellation:** End of block	
G29	**Tridimentional tool correction with 3 or 5 axes** **Syntax:** N.. [D..] [G01] **G29** X.. Y.. Z.. P.. Q.. R.. [I.. J.. K..] [A.. /B.. /C..] **Cancellation:** G40/G41/G42	
G31	**Thread chasing cycle** **Syntax** (XY plane): N.. [G17] [M03/M04] [S..] **G31** [X.. Y..] Z.. [ER..] [EH..] K.. P.. [F..] [EF..] [EC..] **Cancellation:** G80 to G89	

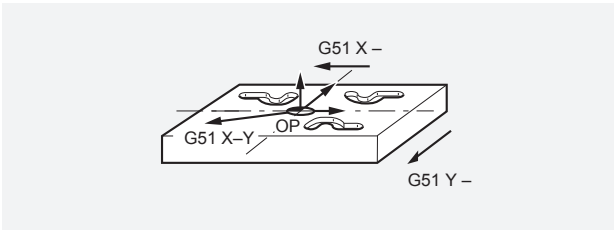
G33	**Constant lead thread cutting** **Syntax:** N.. **G33** X.. Z.. K.. [EA..] [EB..] P.. [Q..] [R..] [F..] [S..] **Cancellation:** End of block	
G38	**Sequenced thread cutting** **Syntax:** N.. **G38** X.. Z.. K.. **Cancellation:** G00/G01/G02/G03	
G38+	**Resume Thread Cutting** **Syntax:** **G38+** J<delta_posX> EC<posB_appris> K<pas> Z<Xfin> **Cancellation:** End of block	
G39+	**Rigid Tap Removal** **Syntax:** **G39+** M3/4 S... K... Z... / **Cancellation:** End of block	
G40	***Radius offset cancel** **Syntax:** N.. [G00/G01] **G40** X.. Y.. Z.. **Cancellation:** G29/G41/G42	
G41	**Left radius offset** **Syntax (XY plane):** N.. [G17] [D..] [G00/G01/G02/G03] **G41** X.. Y.. **Cancellation:** G29/G40/G42	
G42	**Right radius offset** **Syntax (XY plane):** N.. [G17] [D..] [G00/G01/G02/G03] **G42** X.. Y.. **Cancellation:** G29/G40/G41	

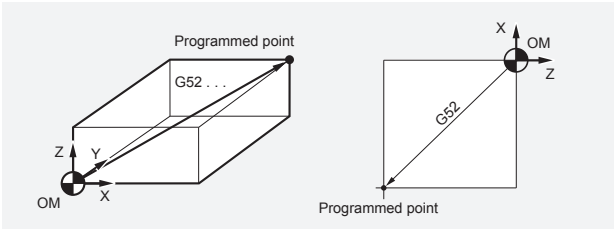
G43	**Tridimensional tool correction with cylindrical tool**
	Syntax: N.. [D..] [G01] **G43** X.. Y.. Z.. P.. Q.. R..[I.. J.. K..] [A.. / B.. / C..]
	Cancellation: G40

G45	**Simple pocket cycle**	
	Syntax: N.. **G45** X.. Y.. Z.. [ER..] EX.. EY.. [EB..] P.. Q.. [I..] [J..] [EG2/EG3] EP.. EQ.. EI.. EJ..	
	Cancellation: End of block	

G48	**Spline curve interpolation**
	Syntax: N.. **G48** NC.. H.. /N.. N..
	Cancellation: End of block

G49	**Spline curve deletion**
	Syntax: N.. **G49** NC..
	Cancellation: End of block

G51	**Mirror function**	
	Syntax: N.. **G51** X- Y- Z-	
	Cancellation: G51 X- Y- Z- cancelled by G51 X+ Y+ Z+	

G52	**Programming movements in absolute dimensions with reference to the measurement origin**	
	Syntax: N.. [G40] [G90] [G00/G01] **G52** X.. Y.. Z.. A.. B.. C.. [F..]	
	Cancellation: End of block	

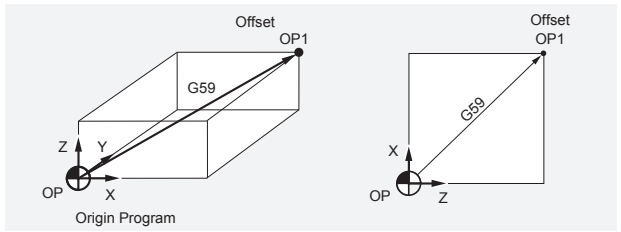
G53	**Cancellation of shifts DAT1 and DAT2**
	Syntax: N.. **G53**
	Cancellation: G54

G54	***Enabling of shifts DAT1 and DAT2**
	Syntax: N.. **G54**
	Cancellation: G53

G59 — Program origin offset

Syntax:
N.. [G90/G91] **G59** X.. Y.. Z.. U.. V.. W.. A.. B.. C.. [I.. J.. K..]

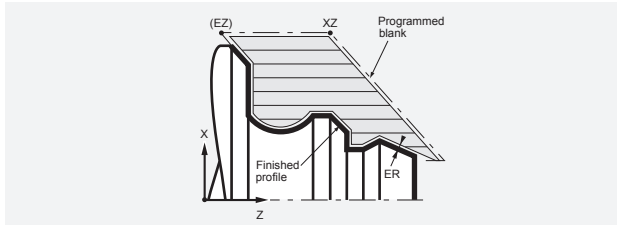
Cancellation:
Cancelled by different G59 X.. Y.. Z..



G63 — Roughing cycle with groove

Syntax:
N.. **G63** N.. N.. X.. Z.. EX.. / EZ.. P.. / R.. EA.. / EU.. EW.. [EB..] [EC..] [ER..] [Q..] [EQ..] [EF..]

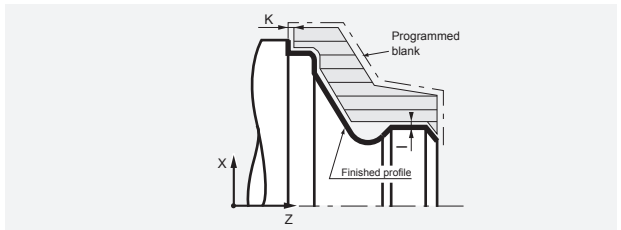
Cancellation:
End of block



G64 — Turn/Face roughing cycle

Syntax:
N.. **G64** N.. N.. [I.. K..] P.. / R..
N.. BLANK DEFINITION

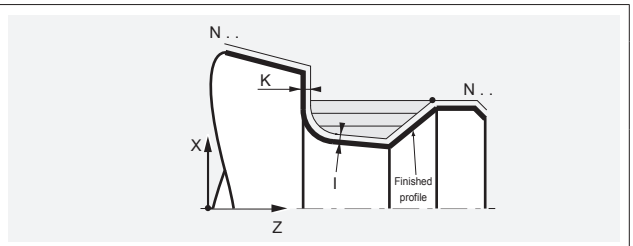
Cancellation:
G80



G65 — Groove roughing cycle

Syntax:
N.. **G65** N.. N.. X.. / Z.. [I.. K..] [EA..] P.. / R.. [Q..] [EF..]

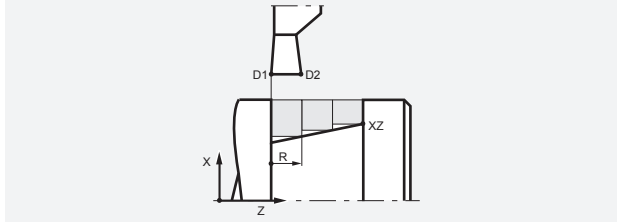
Cancellation:
End of block



G66 — Plunging cycle

Syntax:
N.. **G66** D.. X.. Z.. [EA..] P.. / R.. [EF..]

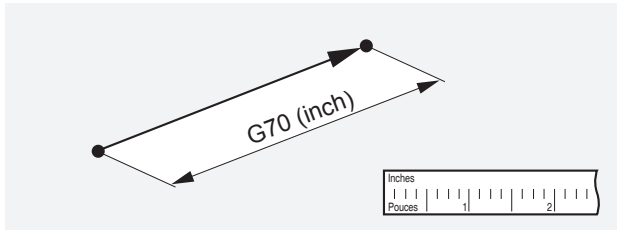
Cancellation:
End of block



G70 — Inch data input

Syntax:
N.. **G70**

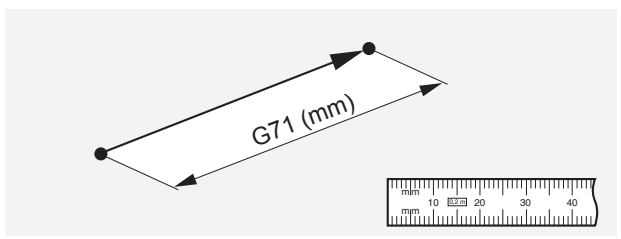
Cancellation:
G71



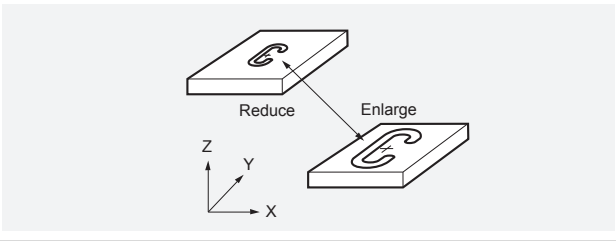
G71 — *Metric data input

Syntax:
N.. **G71**

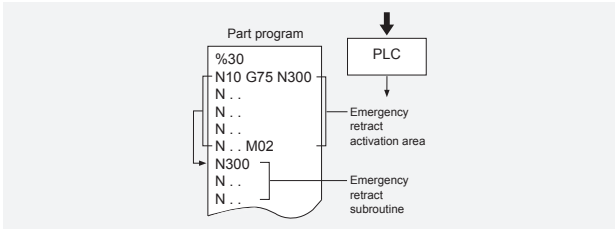
Cancellation:
G70



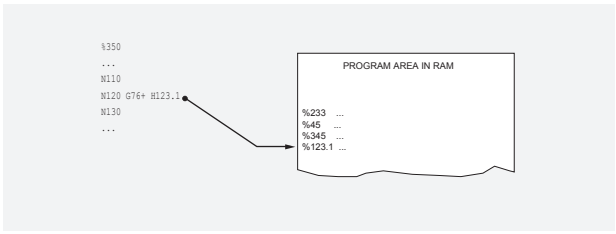
G73	***Scaling factor cancel** **Syntax:** N.. [G40] **G73** **Cancellation:** G74
G74	**Scaling factor enable** **Syntax:** N.. [G40] **G74** **Cancellation:** G73



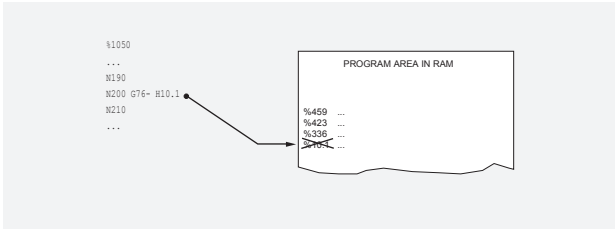
G75	**Emergency retraction subroutine declaration** **Syntax:** N.. **G75** N.. **Cancellation:** Cancelled by G75 N0 or different G75 N..



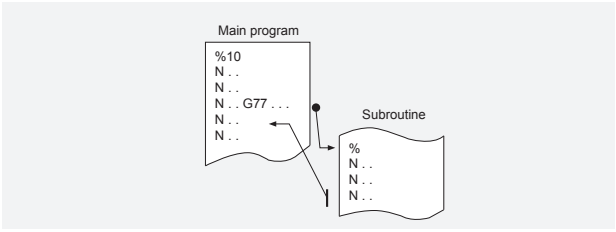
G76	**Transfer of the current values of L and E parameters into the part program** **Syntax:** N.. **G76** [H..] [N.. N..] **Cancellation :** End of block
G76+/-	**ISO program or block creation/deletion** The syntax specific to each function is described below **Cancellation :** End of block
G76+	**Program creation** **Syntax:** N.. **G76+** H.. Note. G76+ H<bin> is used to create binary files. Detailed information can be found in the Flexium Extended Programming manual M00020.

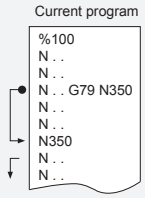
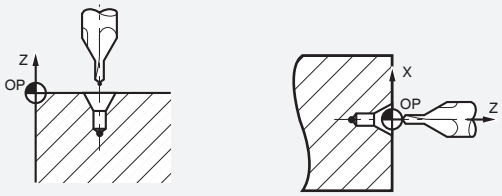
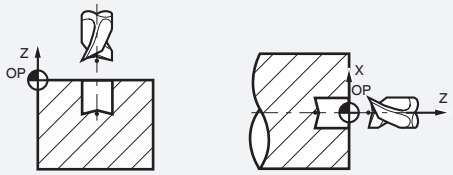
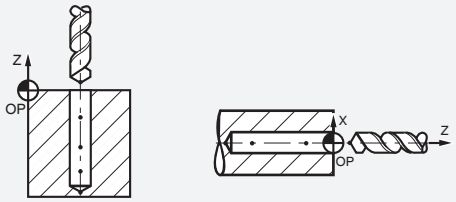


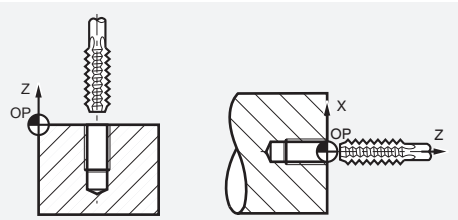
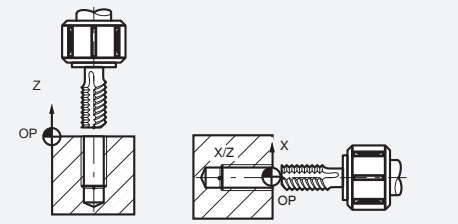
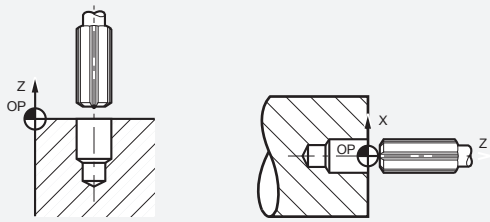
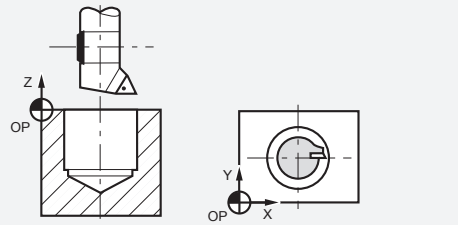
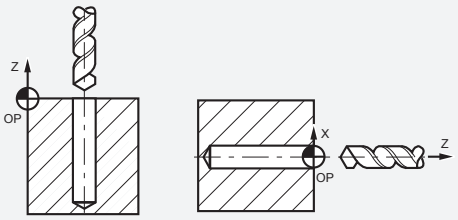
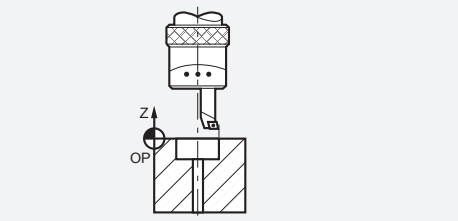
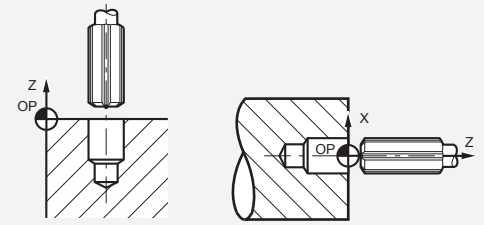
G76-	**Program deletion** **Syntax:** N.. **G76-** H..

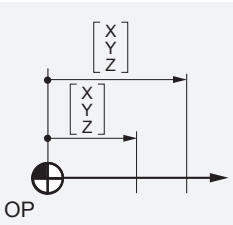
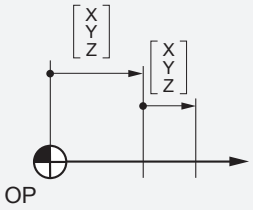
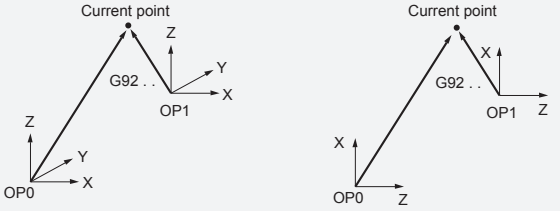
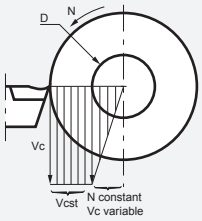
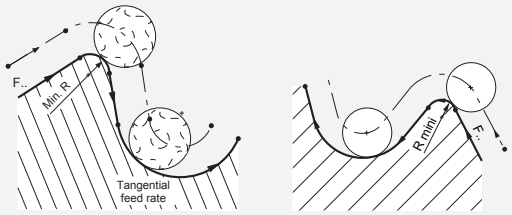
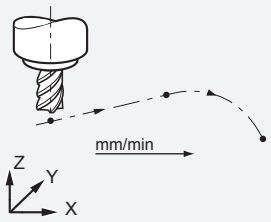
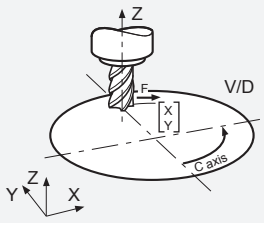


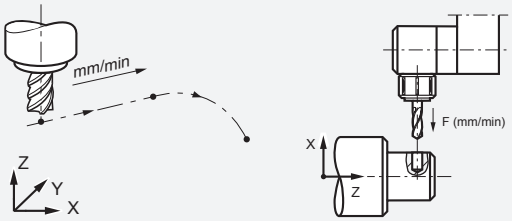
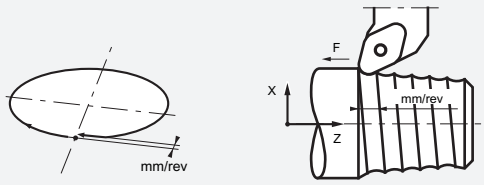
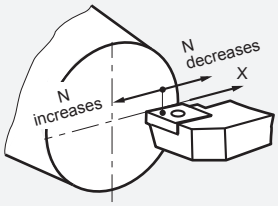
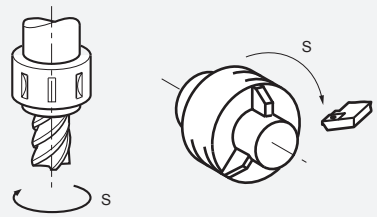
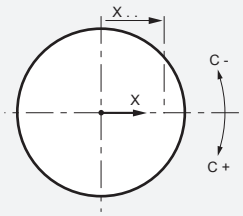
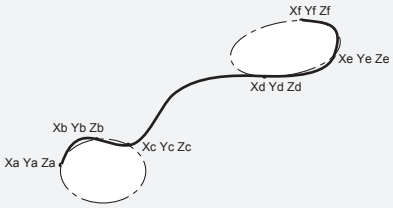
G77	**Unconditional branch to a subroutine or sequence of blocks with return** **Syntax:** N.. **G77** [H..] [N.. N..] [S..] **Cancellation :** End of block



G77 -i	**Call of the subroutine return block**	
	Syntax: N.. **G77 -i**	
	Cancellation : End of block	
G78	**Channel synchronisation with markers**	Please refer to example on page 409.
	Syntax: N.. **G78** Q.. Pj.i Pk.l	
	Cancellation : Other G78 End of program	
G79	**Conditional or unconditional jump to a sequence without return**	
	Syntax: N.. **G79** [L../E.. > = < Number] N..	
	Cancellation : End of block	
G79 +/-	**Temporary suspension of next block preparation in a sequence with movements**	
	Syntax (XY plane) N.. [G00/G01/G02/G03] X.. Y.. Z.. **G79 +/-** X.. /F..	
	Cancellation : End of block	
G80	***Machining cycle cancel**	
	Syntax: N.. **G80**	
	Cancellation: G31/G81-G89	
G81	**Drilling/centre drilling cycle**	
	Syntax (XY plane): N.. [G17] **G81** [X.. Y..] Z.. [ER..] [EH..] [F..]	
	Cancellation: G31/G80/G82-G89	
G82	**Counterboring cycle**	
	Syntax (XY plane): N.. [G17] **G82** [X.. Y..] Z.. [ER..] [EH..] EF.. [F..]	
	Cancellation: G31/G80/G81/G83-G89	
G83	**Peck drilling cycle**	
	Syntax (XY plane): N.. [G17] **G83** [X.. Y..] Z.. [ER..] [EH..] [P..]/[ES..] [Q..] [EP..] [F..] [EF..]	
	Cancellation: G31/G80-G82, G84-G89	

G84	**Tapping cycle** **Syntax (XY plane):** N.. [G17] **G84** [X.. Y..] Z.. [ER..] [EH..] EF.. [F..] **Cancellation:** G31/G80-G83, G85-G89	
G84	**Rigid tapping cycle** **Syntax (XY Plane):** N.. [G17] [M03/M04] [M04-M05] **G84** [X.. Y..] Z.. [ER..] [EH..] K.. [EK..] **Cancellation** G31/G80-G83, G85-G89	
G85	**Reaming cycle** **Syntax (XY plane):** N.. [G17] **G85** [X.. Y..] Z.. [ER..] [EH..] [F..] [EF..] **Cancellation:** G31/G80/G81-G84, G86-G89	
G86	**Boring cycle with indexed stop and clearance at hole bottom** **Syntax (XY plane):** N.. [G17] **G86** [X.. Y..] Z.. [ER..] [EH..] [EC..] [EA..] [EP..] [F..] **Cancellation:** G31/G80-G85, G87-G89	
G87	**Drilling cycle with chip breaking** **Syntax (XY plane):** N.. [G17] **G87** [X.. Y..] Z.. [ER..] [EH..] [P..]/[ES..] [Q..] [EP..] [EF..] [F..] **Cancellation:** G31/G80-G86, G88/G89	
G88	**Boring and facing cycle** **Syntax (XY plane):** N.. [G17] **G88** [X.. Y..] Z.. [ER..] [EH..] [F..] **Cancellation:** G31/G80-G87/G89	
G89	**Boring cycle with Dwell at the Bottom of the hole** **Syntax (XY plane):** N.. [G17] **G89** [X.. Y..] Z.. [ER..] [EH..] [EF..] [F..] **Cancellation:** G31/G80-G88	

G90	***Absolute dimensions with respect to the program origin** **Syntax:** N.. **G90** X.. Y.. Z.. A.. B.. C.. **Cancellation:** G91	
G91	**Incremental dimensions with respect to the start of the block** **Syntax:** N.. **G91** X.. Y.. Z.. A.. B.. C.. **Cancellation:** G90	
G92	**Program origin preset** **Syntax:** N.. **G92** X.. Y.. Z.. **Cancellation:** End of block	
G92 S..	**Spindle Speed Limiting** N.. **G92** S.. **Cancellation:** Cancelled by G92 S0 or G92 with different S.	
G92 R..	**Programming the tangential feed rate** **Syntax:** N.. **G92** R.. **Cancellation:** Cancelled by G92 R0 or by different G92 R	
G92 F	**Setting feed rate per axis** **Syntax:** **G92F...** <list of axes> **Cancellation:** G92 F... or M02	
G93	**Feed rate in inverted of time (V/L)** **Syntax:** N.. **G93** F.. G01 X.. Y.. Z.. A.. B.. C.. **Cancellation:** G94/G95	

G94	***Feed rate expressed in millimetres, inches or degrees per minute** **Syntax:** N.. **G94** F.. G01/G02/G03 X.. Y.. Z.. A.. B.. C.. **Cancellation:** G93/G95	
G95	**Feed rate expressed in millimetres or inches per revolution** **Syntax:** N.. **G95 F**.. G01/G02/G03 X.. Y.. Z.. **Cancellation:** G93/G94	
G96	**Constant surface speed is expressed in meters per minute** **Syntax:** N.. **G96** [X..] S.. **Cancellation:** G94	
G97	***Spindle speed Control (RPM)** **Syntax:** N.. **G97 S**.. [M03/M04] **Cancellation:** G96 (combined machine)	
G98	**Definition of the Start X for Interpolation on the C Axis** **Syntax:** N.. **G98** X.. **Cancellation:** End of block	
G104	**Tridimensional curve smoothing** **General syntax:** N.. X. . Y.. Z.. (first point on the curve) N.. [G01] **G104** X..Y.. Z.. [F..] [Intermediate points on the curve] N..G80 X.. Y.. Z.. (last point on the curve) **Cancellation:** G80	
G153 to G159	**Functions reserved for measuring cycles** These functions are parameterised by the machine builder. Please refer to the measuring cycles documentation.	

G199	**Programmed homing**
	Syntax: **G199** <list of axes to home>
	Cancellation: End of block Remark This function is parameterised by the machine builder. Refer to the machine user's documentation.

G997	**Enabling and execution of all functions stored in state G999**
	Syntax: N.. **G997**
	Cancellation: G998/G999

G998	**Enabling of execution of the blocks and part of the functions processed in state G999**
	Syntax: N.. **G998**
	Cancellation: G997/G999

G999	**Suspension of execution and forcing of block concatenation**
	Syntax: N.. **G999**
	Cancellation: G997/G998

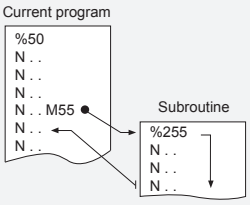
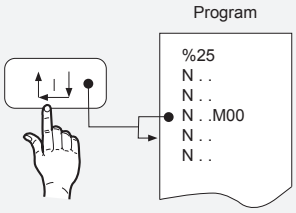
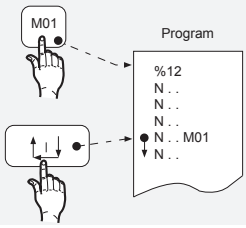
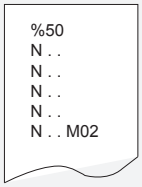
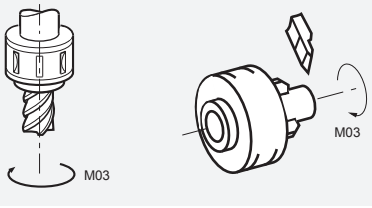
5.1.2 M Functions - Main summary table

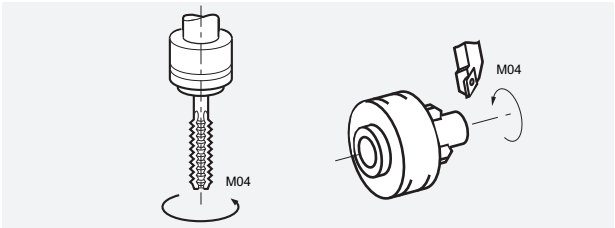
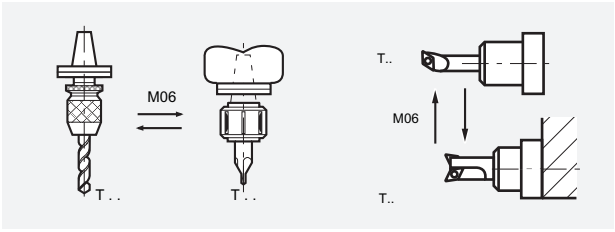
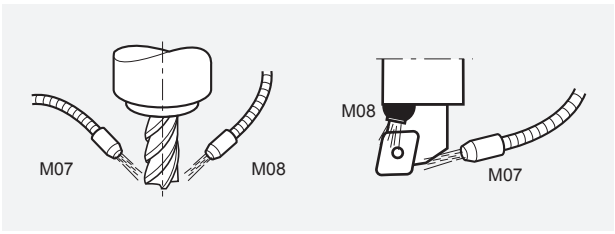
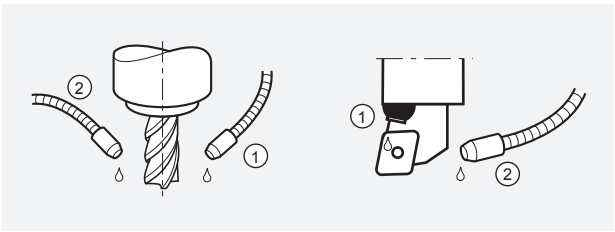
The functions initialized at power on are identified by «*».

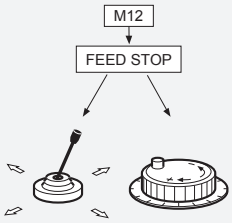
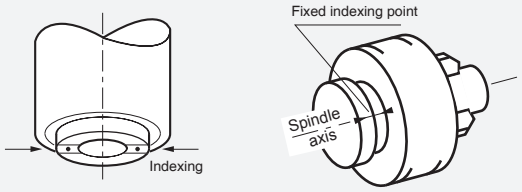
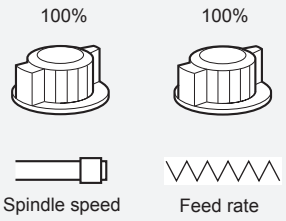
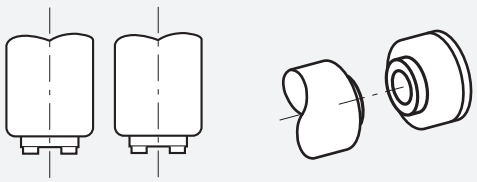
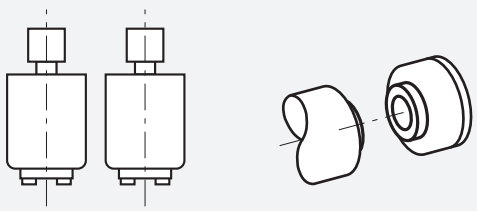
The miscellaneous functions listed in the table are decoded functions. Several decoded M functions can be programmed in the same block, e.g. N.. S100 M03 M40 M08.

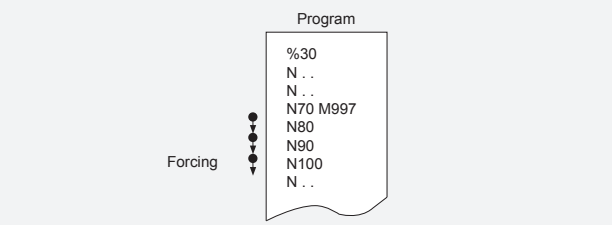
Click on **Mxx** to give the detailed information.

The "Syntax" indications in the table are given for Milling application only.
Please refer to the detailed Mxx informations to give the "Syntax" on Turning application.

M	**Subroutine Branch by M Function** N.. **M..**	
M00	**Program stop** **Syntax**: N.. [G40] **M00** [$0 ...] **Cancellation**: Action on the machine panel CYCLE key	
M01	**Optional stop** **Syntax**: N.. [G40] **M01** [$0 ...] **Cancellation**: Action on the machine panel CYCLE key	
M02	**End of program** **Syntax**: N.. **M02**	
M03	**Spindle clockwise rotation** **Syntax**: N.. **M03** **Cancellation**: M04/M05/M00/M19	

M04	**Spindle counterclockwise rotation** **Syntax**: N.. **M04** **Cancellation**: M03/M05/M00/M19	
M05	***Spindle off** **Syntax**: N.. **M05** **Cancellation**: M03/M04	
M06	**Tool change** **Syntax**: N.. T.. [D..] **M06** [$0.. / (...)] **Cancellation**: M function report (CRM).	
M07	**Coolant 2 on** **Syntax**: N.. **M07** **Cancellation**: M09.	
M08	**Coolant 1 on** **Syntax**: N.. **M08** **Cancellation**: M09	See figure above.
M09	***Coolant off** **Syntax**: N.. **M09** **Cancellation**: M07/M08	
M10	**Clamp** **Syntax**: N.. [G00/G01/ G02/ G03] **M10** X.. Y.. Z.. A.. B..C.. **Cancellation**: M11	
M11	**Unclamp** **Syntax**: N.. **M11** **Cancellation**: M10	

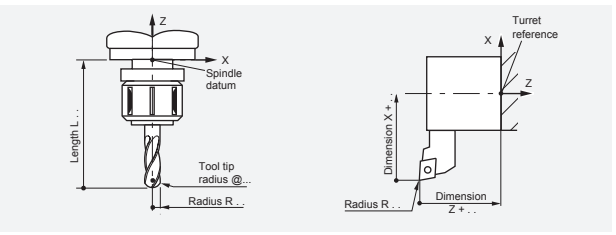
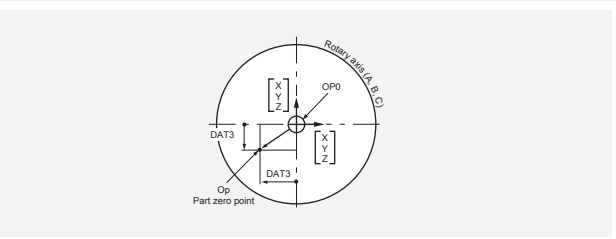
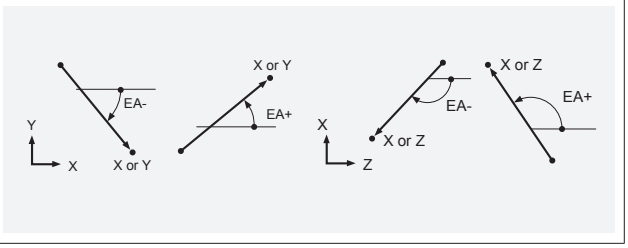
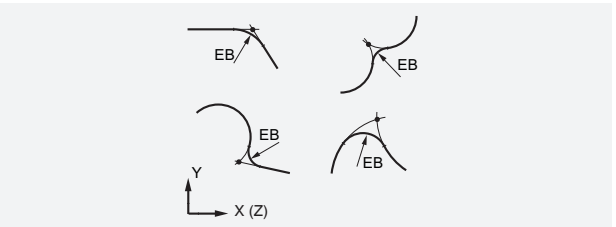
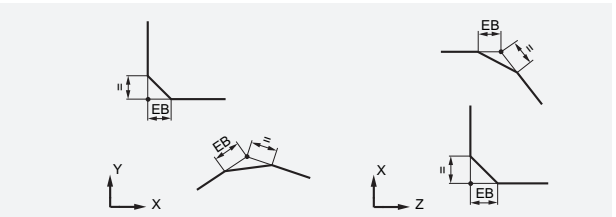
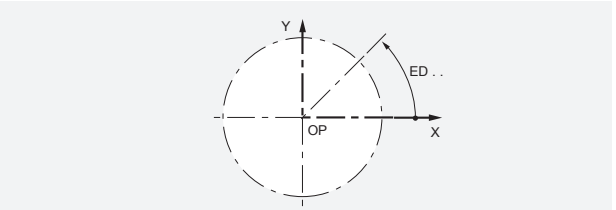
M12	**Programmed feed stop** **Syntax**: N.. **M12** [$0...] **Cancellation**: Press CYCLE on the machine panel	
M19	**Indexed Spindle Stop** **Syntax**: N.. [S..] [M03/M04] EC±.. **M19** **Cancellation**: M03/M04/M05.	
M40 **M41** **M42** **M43** **M44** **M45**	**Spindle speed ranges** **Syntax**: N.. [S..] [M03/M04] **M40 to M45** **Cancellation**: Cancel one another	
M48	***Spindle speed and feed rate potentiometer enable** **Syntax**: N.. **M48** **Cancellation**: M49	
M49	**Spindle speed and feed rate potentiometer enable inhibit** **Syntax**: N.. **M49** **Cancellation**: M48	See figure above.
M61	**Release the current spindle in the channel** **Syntax**: N.. **M61** **Cancellation**: M62-M65	
M64 **M65** **M62** **M63**	**Control of spindles 1 to 4 (*M64)** **Syntax**: N.. [S..] **M62/M63/M64/M65** [M40-M45] M03/M04 **Cancellation**: Cancel one another	
M66 **M67** **M68** **M69**	**Spindle 1 to 4 measurement (*M66)** **Syntax**: N.. [S..] **M66/M67/M68/M69** **Cancellation**: Cancel one another.	

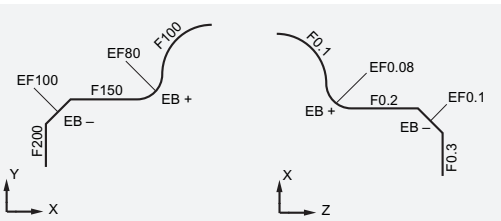
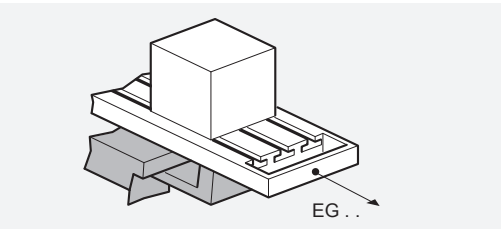
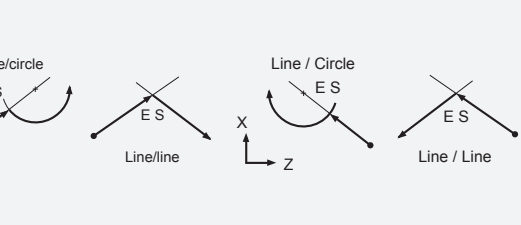
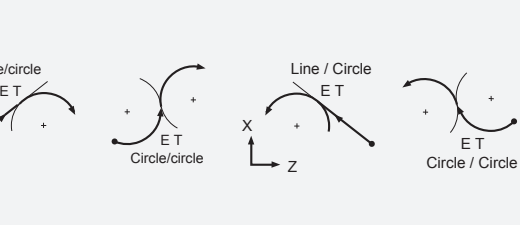
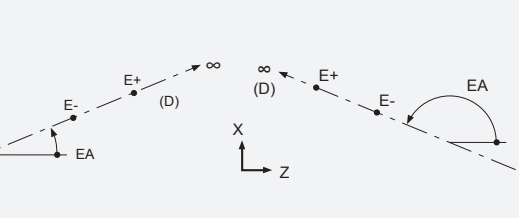
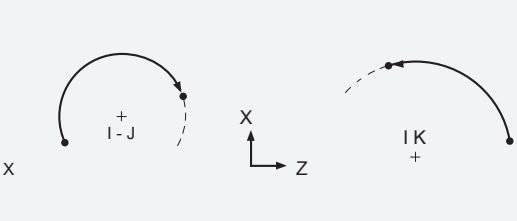
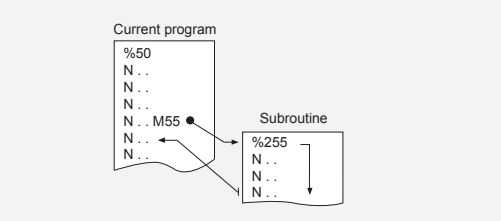
M997	**Forced block consecutive execution**	
	Syntax: N.. **M997**	
	Cancellation: M998/M999/M02.	

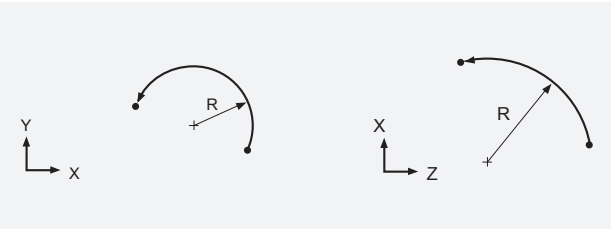
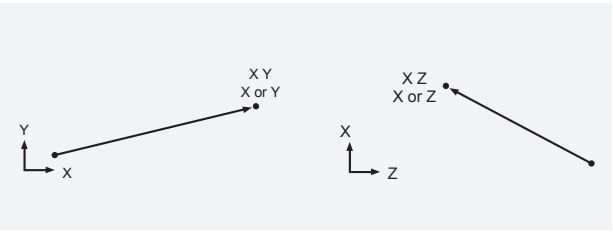
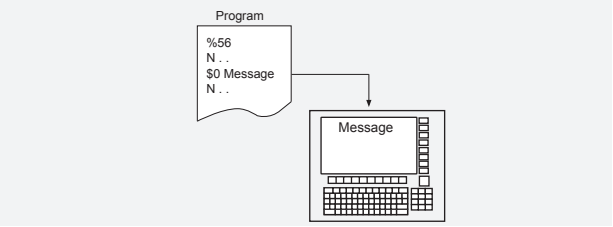
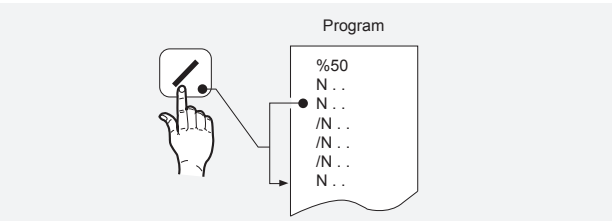
M998	***Reactivation of MDI mode and subroutine calls by the automatic control function**
	Syntax: N.. **M998**
	Cancellation: M997/M999

M999	**Programmed cancellation of MDI mode and subroutine calls by the automatic control function**
	Syntax: N.. **M999**
	Cancellation: M997/M998/M02

5.1.3 Other Functions - Main summary table

D..	**Tool Offset activation** **Syntax**: N.. [G17] [G16 R+] **D..** [G40/G41/G42] X.. Y.. Z.. **Cancellation**: D0 , M02 or new tool offset	
DAT3	**Milling Table offset**	
EA..	**Angle element of a line**	
EB+	**Fillet between two interpolations** **Syntax**: N.. G01/G02/G03 X.. Y.. I.. J.. / R.. [F..] **EB+**.. [EF..] **Cancellation:** End of block	
EB-	**Chamfer between two linear interpolations** **Syntax**: N.. G01 X.. Y.. [F..] **EB-**.. [EF..] **Cancellation:** End of block	
ED..	**Programmed angular offset** **Syntax**: N.. [G90/G91] **ED..** **Cancellation**: Programming of a new value, M02, reset.	

EF	**Feed rate specific to fillets EB+ and chamfers EB-.** **Syntax**: N.. Interpolation EB+.. / EB-.. **EF..** **Cancellation**: Programming of a new value, M02, reset.	
EG..	**Programmed acceleration reduction** **Syntax**: N.. **EG..** **Cancellation**: Programming of a new value, M02, reset.	
ES	**Secant element**	
ET	**Tangent element**	
E+/E-	**Discriminant**	
I.. J.. **K..**	**Coordinates of the centre of a circle**	
M..	**Subroutine branch by M function** **Syntax**: N.. **M..**	

R..	**Radius of a circle**	
X../Y.., Z.. or X.. Y.., Z..	**Coordinates of the end point of a line**	
$0	**Message Transmission to the display** **Syntax**: N.. **EG..** **Cancellation**: Programming end program, M02, reset.	
/	**Block skip** **Syntax**: / N.. (Contents unimportant)	

5.1.4 Word Format - Main summary table

The formats of the axis words specified in the table are expressed as follows:

- For linear axes: 5 digits to the left and 3 to the right of the decimal point are allowed when the internal system unit (see chapter 3.1) is µm.
- For rotary axes: 3 digits to the left and 4 to the right of the decimal point are allowed when the internal system unit is 0.0001 degree.

For instance, for linear axes: if the system is set to 0.1µ (internal unit), the formats are expressed with 4 decimal digits; for the X axis, the format is X+044.

For instance, for rotary axes: if the system is set to 0.001 degree (internal unit), the formats are expressed with 3 decimal digits; for the B axis, the format is B+033.

For the words related to the machining feed rate with no assigned formats (F.., EF..), refer to the machine manufacturer's manual for the maximum and minimum feed rates (maximum 8 digits and decimal point).

Format	Description
%051	Program number (1 to 99999.9)
N05	Sequence number (1 to 2147483647)
G02 **G03**	Preparatory functions (0 to 99) Preparatory functions (100 to 250 and 997 to 999)
H05	Subroutine number (with G77, G76 and G48)
X+053	Movement on the X axis. In a cycle, end point on the machining axis (.. plane)
Y+053	Movement on the Y axis. In a cycle, end point on the machining axis (.. plane)
Z+053	Movement on the Z axis. In a cycle, end point on the machining axis (.. plane)
I+053 **J+053** **K+053**	For circular/helical interpolation (G02, G03), centre of the circle and pitch of the helix With programmed origin offset (G59), centre of rotation and angular offset (ED) For thread chasing cycle (G31), thread pitch = I, J or K depending on the interpolation plane
I053	For pocket cycle (G45), axial finishing pass
J053	For pocket cycle (G45), lateral finishing pass
K053	For leadscrew tapping cycle (G84), tap pitch
U+053	Movement on the U axis
V+053	Movement on the V axis
W+053	Movement on the W axis
A+034	Movement on the A axis
B+034	Movement on the B axis
C+034	Movement on the C axis
E+/E-	For profile geometry programming (PGP), discriminant
EA+033 **EA+033** **EA+033**	For profile geometry programming (PGP), line angle In a boring cycle with indexed spindle stop (G86), angle between EC.. and tool cutting edge In polar programming, angle of the line
EB+053 **EB-053** **EB053**	For contour definition and PGP, radius or fillet between two interpolations For contour definition and PGP, chamfer between two linear interpolations For profile Geometry Programming (PGP), chamfer between 2 lines
EC+033 **EC+033** **EC+033**	For spindle indexing (M19), indexing angle For thread chasing cycle (G31), tool orientation in the hole bottom For boring cycles (G86), indexing position
ED+034	Programmed angular offset
EF022 **EF..** **EF..** **EF..**	For cycles (G82, G84, G87, G89 or G31), dwell time In a reaming cycle (G85), retraction rate Feed rate specific to fillets (EB+) or chamfers (EB-) Maximum feed rate after an interrpt (G10)
EG01	Pocket cutting direction (EG02: clockwise; EG03: counterclockwise)

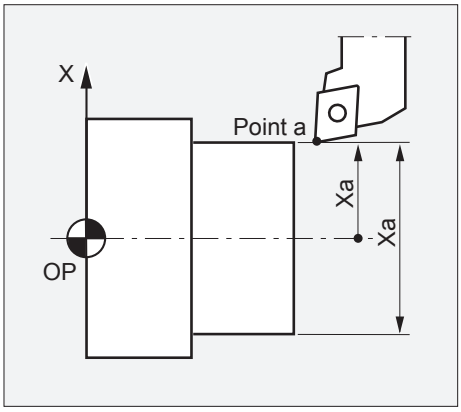
EG03	For interpolation, acceleration modulation
EH+053	In (G8x) cycles, dimension of the impact plane on the machining axis
EI+053 **EI..**	In polar programming, length of the line (start/circle center) In a pocket cycle (G45), axial finishing feed rate
EJ..	In a pocket cycle (G45), lateral finishing feed rate
EK01	For rigid tapping cycle (G84), retraction/penetration speed ratio
ES+/ES-	For profile geometry programming (PGP), secant element
ET+/ET-	For profile geometry programming (PGP), tangent element
EP053 **EP053** **EP053** **EP..**	In a boring cycle with indexed spindle stop (G86), lateral backoff in hole bottom In a peck drilling cycle (G83), backoff clearance after each peck In a drilling cycle with chip breaking (G87), backoff between two infeeds In a pocket cycle (G45), axial roughing feed rate
EQ053	For pocket cycle (G45), lateral roughing feed rate
ER+053	In cycles, infeed or retraction dimension on the machining axis
ES02	In cycles G83 and G87, number of infeeds at constant value
EX+053 **EX053**	In polar programming, angle of the line (start/end) For pocket cycle (G45), dimension along X (or U)
EY053	For pocket cycle (G45), dimension along Y (or V)
EZ053	For pocket cycle (G45), dimension along Z (or W)
P+053 **P+043** **P053** **P043** **P041** **P+ / P-**	For thread chasing cycles (G31), withdrawal of tool For 3D correction (G29), X component of the normal vector For cycle (G83 or G87), value of the first penetration For pocket cycles (G45), axial roughing path With axis channel synchronisation (G78), wait for a marker Tool axis orientation (G16), along X (or U)
Q+053 **Q+043** **Q043** **Q04** **Q+ / Q-**	For cycles (G83 or G87), value of the last penetration For correction in space (G29), Y component of the normal vector For pocket cycle (G45), lateral roughing pass With channel synchronisation (G78), declaration of a marker Tool axis orientation (G16), along Y or V
R+053 **R053** **R+043** **R+ / R-** **R+ / R-**	For circular/helical interpolation (G02, G03), radius of the circle For tangential feed rate programming (G92), value of the curve radius For space correction (G29), Z component of the normal vector For linear interpolation (G00, G01), positioning at programmed distance Tool axis orientation (G16), along Z or W
F.. **F..** **F..** **F..** **F..** **F022** **F01**	Feed rate in mm/min and degrees/min (G94) Feed rate in inches/min (G94) Feed rate in mm/rev (G95) Feed rate in V/L (G93) Feed rate in inches/revolution (G95) Dwell in seconds (G04) For thread chasing cycles (G31), number of threads (1 to 9)
M02 **M03**	Miscellaneous functions 0 to 99 Miscellaneous functions 100 to 899
NC04	For spline curve interpolation (with G06, G48, G49), curve number (1 to 9999)
S05 or S032 **S02**	Spindle rotation speed in rpm (G97) In a subroutine branch (G77), number of subroutine iterations (1 to 99)
T08	Tool number (0 to 99999999)
D03	Correction number (0 to 255)
L03	Program variables L (0 to 19, 100 to 199 and 900 to 959)
E5	External parameters E

5.2 Programming with Reference to Diameter or Radius

The part program and certain data related to machining along the X (or U) axis are directly affected by whether programming is with reference to diameter or radius.

Programming with reference to diameter or radius is selected by the machine builder.

In both cases, certain functions are always expressed with respect to diameter and others with respect to radius.



5.2.1 System Programmed with Reference to Diameter

Programmed values expressed with reference to diameter:
- values programmed in absolute dimensions (G90): coordinates of a movement along X.. and position I.. of the centre of a circle,
- value of the starting diameter with constant surface speed (G96),
- value programmed with function G98.

Programmed values expressed with reference to radius:
- values programmed in incremental dimensions (G91): value of a movement along X.. and position I.. of the centre of a circle,
- circle radius with circular interpolation (R),
- fillet or chamfer (EB+, EB-),
- pass depth for roughing cycle (P or R),
- machining allowance for roughing cycles (I or K), (ER),
- positioning clearance for roughing cycle (Q),
- minimum depth of cut for roughing cycle (EQ),
- thread depth (P) and last pass for thread cutting (Q),
- pass depth for drilling cycle (P and Q),
- offsets programmed with functions G59 and G52.

Tool dimensions:
- Entry of values with respect to radius.

Dynamic tool corrections:
- Entry of values with respect to diameter, but display of radial change on the «context "Tools"» page.

Value of offset DAT2:
- Entry of the value with reference to radius.

Movements related to the manual controls:
- Movements on the X axis with reference to radius, but display with reference to diameter on the «context "production"» page.

5.2.2 System Programmed with Reference to Radius

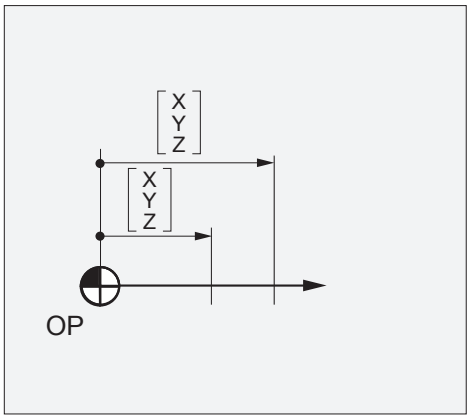
Values expressed with reference to radius:
- All the programmed movements applied to the X axis and all the values entered related to machining along X.

Values expressed with reference to diameter:
- Only the dynamic tool corrections on X are entered with reference to diameter.

5.3 Absolute or Incremental selection ► G90, G91

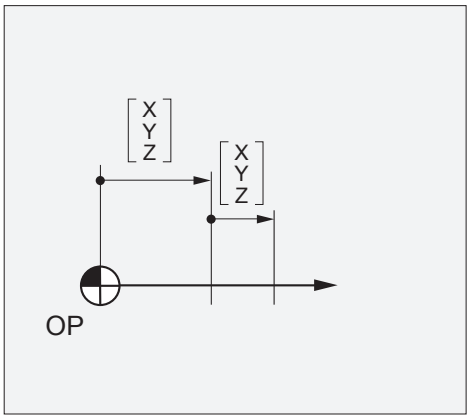
G90	Absolute dimensions with respect to the program origin.

The value programmed on an axis is referenced to the program origin (OP).



G91	Incremental dimensions with respect to the start of the block.

The value programmed on an axis is referenced to the last programmed position. The value is equal to the movement to be performed.



Syntax

N.. **G90/G91** X.. Y.. Z.. A.. B.. C..

G90	Absolute dimensions.
G91	Incremental dimensions.
X.. Y.. Z.. A.. B.. C..	End points.

Properties of the Functions

Functions G90 and G91 are modal.

G90 is the default function.

Cancellation

Functions G90 and G91 cancel one another.

Important note

The first movement must be programmed:

- in absolute dimensions (G90) either in MDI or in the part program.
- if a first movement is programmed in incremental, it will be considered as an absolute positioning anyway. No error message will be generated.
- incremental programming (G91) is prohibited for PGP (Profile Geometry Programming, see Chapter 6).

Combined programming

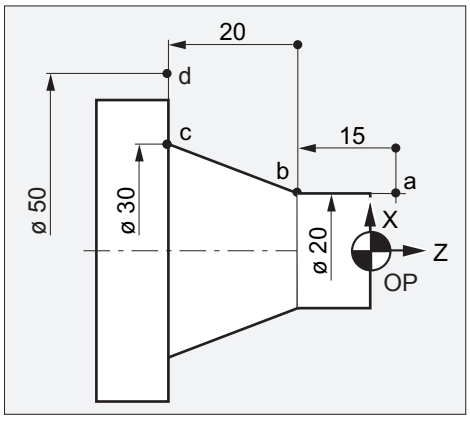
Both types of programming (G90/G91) can be included in a program and even in a block.
For instance:

```
N..
N.. G91 X.. Y..
N.. G90 X.. G91 Y..                    X absolute, Y incremental
N.. G90 X.. Y..
N..
```

Example

Incremental programming (G91), (System programmed on diameter)

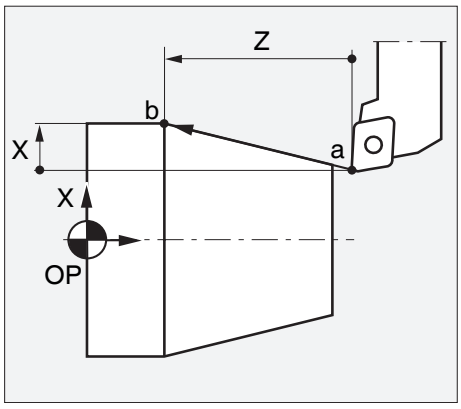
Absolute programming of point a,
incremental movement to points b, c, d.




```
N.. (G90) ...
N.. X20 Z5
N.. G91 Z-15
N.. X5 Z-20
N.. X10
N.. G90
N..
```

Incremental programming (G91), (System programmed on radius)

Tool located at point a (starting point)
Incremental programming of point b
(amount of movement to reach end point b).

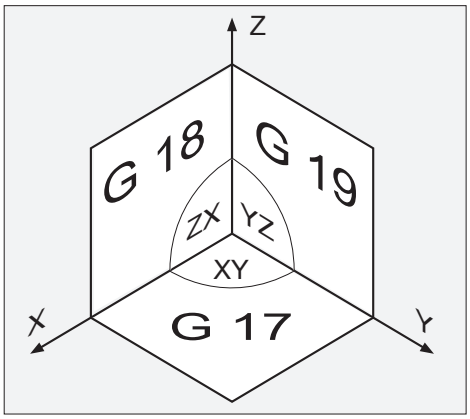



```
N.. (G90) ...
N.. Xa Za
N.. G91 Xb Zb
N..
```

5.4 Plane Selection ► G17, G18, G19

G17/G18/G19 Plane selection for circular interpolation and radius correction.

One of these G functions is programmed to define the radius correction and circular interpolation plane.



Syntax

N.. **G17/G18/G19**

G17 XY plane selection.

G18 ZX plane selection.

G19 YZ plane selection.

Properties of the Functions

Functions G17, G18 and G19 are modal.

G17 is the default function.

Cancellation

Functions G17, G18 and G19 cancel one another.

Notes

A change of plane must be programmed:

- with the system in state G40 (no G41/G42 cutter compensation active) or the system returns error message 138.
- after a sequence completely defined in «PGP» (Profile Geometry Programming, see Chapter 6) or the system returns error message 137.

Example

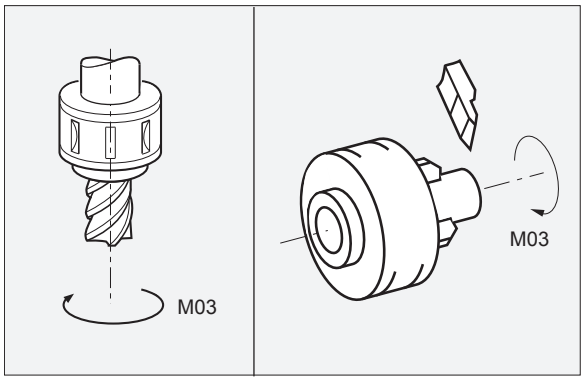
```
N.. ...
N50 G17...                    XY plane selection
N..
N160 G18 ...                  ZX plane selection
N..
N250 G17 ...                  XY plane selection
N..
```

5.5 Spindle Control

5.5.1 Spindle Direction of Rotation ► M03, M04, M05

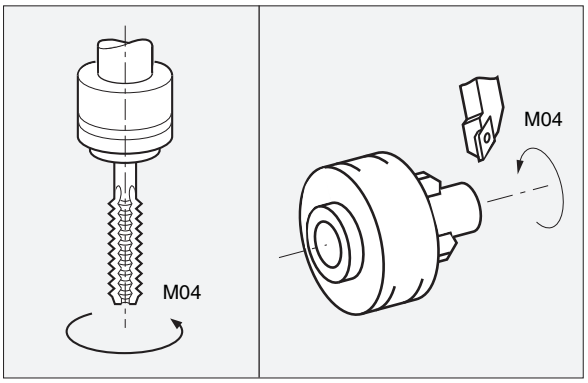
M03	Spindle clockwise rotation.

This command starts spindle rotation at the speed programmed.



M04	Spindle counterclockwise rotation.

This command starts spindle rotation at the speed programmed.



M05	Spindle off.

This command stops spindle rotation.

Syntax

N.. **M03/M04/M05**

M03 Spindle clockwise rotation.

M04 Spindle counterclockwise rotation.

M05 Spindle off.

Properties of the Functions

Functions M03 and M04 are decoded modal «pre» functions.

Function M05 is a decoded modal «post» function. It is the default function.

Cancellation

Functions M03, M04 and M05 cancel one another.

Functions M00, M19 and M01 (enabled) cancel functions M03 or M04.

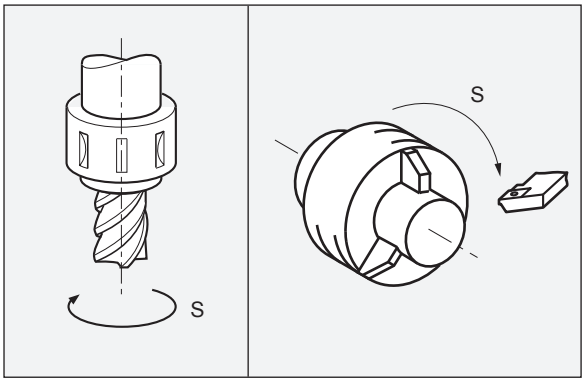
Example

```
N.. ...
N120 ...(MILLING CUTTER WITH R/H HELIX)   Tool call
N130 M03 ...                              Spindle clockwise rotation
N..
N..
N220 M05 ...                              Spindle off
N..
```

5.5.2 Spindle Speed Control ► G97

G97	RPM spindle speed.

This function defines a fixed spindle speed programmed with the S word.



Syntax

N.. **G97** S.. [M03/M04]

G97	Function setting the spindle speed in rpm.
S..	Mandatory argument associated with the function to define the speed.
M03/M04	Spindle direction of rotation.

Properties of the Functions

Function G97 is modal. It is the default function.

Cancellation

On Milling Machine function G97 is the single operating mode.

On Turning this function is cancelled by G96 S...
(constant surface speed).

The spindle speed programmed by G97 is cancelled by S0 or modified by programming S.. followed by a new value.

Notes

Spindle speed format

The speed format may differ according to the machine type:
- Format S05 (1 to 65000 rpm),
- Format S032 (0.01 to 650 rpm).

Example

```
N.. ...
N130 G97 S636 M03                    Spindle rotation
N..
```

Reminder

The spindle rotation speed N is determined from the required cutting speed (V).

The cutting speed V in metres per minute is mainly related to:

- the tool material,
- the part material.
- Cutting speed V = 20 m/min.
- Tool diameter D = 10 mm.

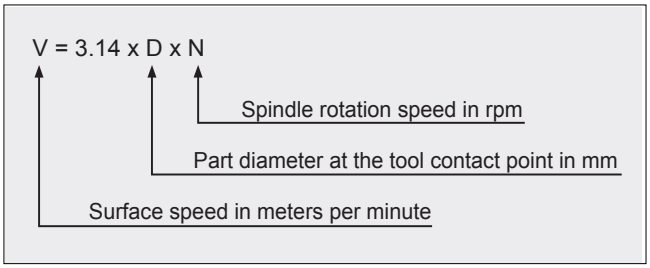
$$N\ (rpm) = \frac{V}{3.14 \times D}$$

$$N = \frac{20000}{3.14 \times 10}$$

N = 636.9 rpm, i.e. S636

5.5.2.1 Review of surface speed

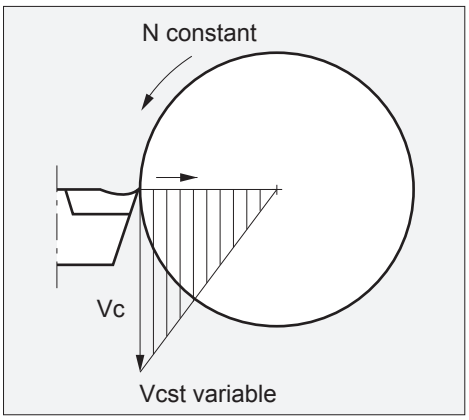
General equation for surface speed V:



5.5.2.2 Using a fixed spindle rotation speed (N)

The fixed spindle speed (N) is expressed in revolutions per minute.

The surface speed V decreases as the tool moves toward the centre of the part, (the surface speed is zero at the centre).

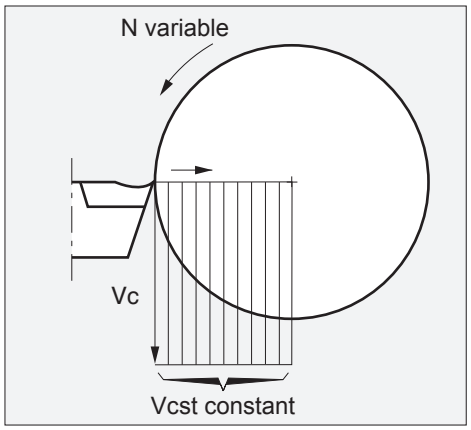


5.5.2.3 Using a constant surface speed (V)

The constant surface speed (Vcst) is expressed in meters per minute.

The surface speed is kept constant as the tool moves toward the centre of the part.

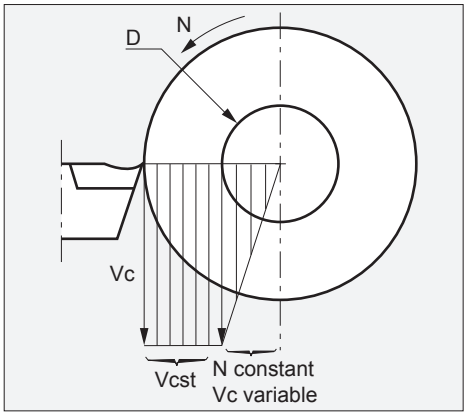
When the tool tip is at the centre, diameter D is equal to zero and speed N should be infinite (physically impossible because of the maximum physical rotation speed of the spindle).



5.5.2.4 Spindle speed limit

A safety limit can be defined by programming a maximum speed in revolutions per minute.

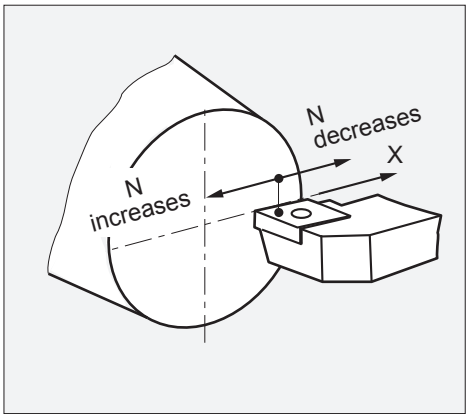
The limit implies a diameter D beyond which a constant surface speed no longer applies. The system returns to the case of a constant rotation speed and a variable surface speed.



5.5.3 Constant Surface Speed on Turning machine ► G96

G96	Constant surface speed is expressed in meters per minute.

This function varies the spindle rotation speed (N).
The rotation speed varies according to the position of the tool centre with reference to the part diameter.



Syntax

N.. **G96** [X..] S..

G96	Function forcing a constant surface speed in m/min.
X..	Argument defining the current diameter.
S..	Mandatory argument associated with the function to define the programmed speed.

Property of the Function
Function G96 is modal.

Cancellation
Function G96 is cancelled by function G97 S..

Notes
This function can only be programmed when the machine is equipped with a variable speed spindle.
The machine spindle must be rotating when the function is called.
After a retraction with reference to the measurement origin (G52 X..), a new constant surface speed initialisation position in X (or U) must be reprogrammed.

When a constant surface speed is programmed:
- the X (or U) axis defining computation of the constant surface speed can be programmed in the same block or one of the previous blocks (between G52 and G96). If X is missing, the system returns error message 28.
- the X (or U) axis must be programmed with reference to the program origin,
- the speed applies to the centre of the tool insert radius,
- it can be modified during the program by redefining a new speed by G96 S..,
- enabling of offsets DAT1 and DAT2 does not affect the surface speed.

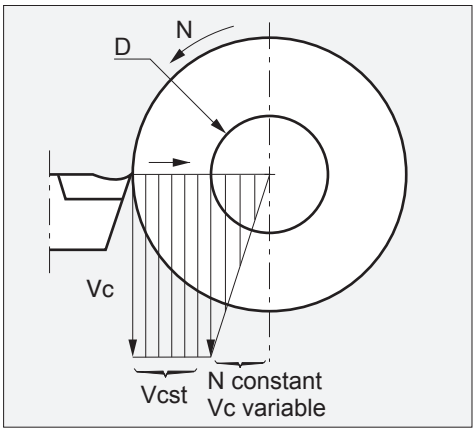
During machining with constant surface speed, it is recommended to:
- program the feed rate in mm/revolution in order to perform machining with a constant chip thickness,
- cancel the constant surface speed by programming a rotation speed in rpm (G97 S..) before each tool change, then initialise the constant surface speed on the X position of the new tool (and D correction). Program a spindle speed limit to prevent danger when the turned diameters decreased below a certain value.

5.5.4 Spindle Speed Limiting ► G92 S

G92 S.. Spindle speed limiting.

This function specifies the maximum spindle speed not to be exceeded.

When facing or parting off at constant surface speed (G96), the rotation speed can be limited to prevent problems due to centrifugal force, imbalance, etc.



Syntax

N.. **G92** S..

G92	RPM spindle speed limiting.
S..	Mandatory argument associated with the function to define the maximum spindle speed.

Property of the Function
Function G92 is modal.

Cancellation
Spindle speed limiting is cancelled by:
- cancellation function G92 S0,
- function G92 S.. with assignment of a different speed limit,
- the end of program function (M02),
- a reset.

Notes
Spindle speed limiting:
- must be programmed before selecting constant surface speed (G96),
- must be programmed at a value below the maximum spindle speed,
- is independent of the maximum speed defined in one of the spindle speed ranges (M40-M45).

If the spindle speed programmed by G97 is higher than the speed defined by G92, the spindle rotates at the limited speed.
G92 must be immediately followed by its argument S...

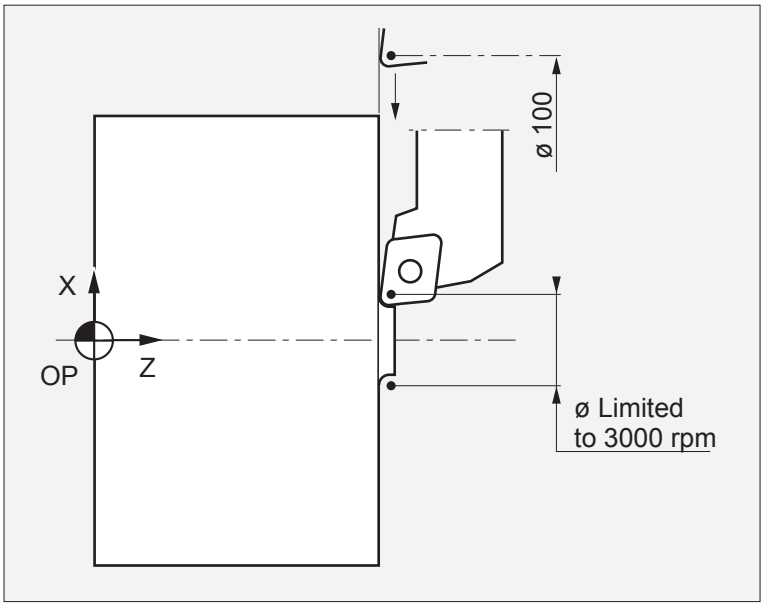
If they are separated by an axis (X.., Z.., etc.), the system interprets function G92 as preselection of the program origin on the programmed axis (see Sec. 5.16.3).

Example

Programming speed limiting for facing at constant surface speed

On read of block N60, the spindle is initialised at 637 rpm on diameter 100 mm.

On execution of block N70, the rotation speed is gradually increased up to a maximum of 3000 rpm (approximately 21 mm diameter). The rest of facing up to X0 is carried out at a speed of 3000 rpm.




```
N.. ...
N30 G97 S900 M40 M04          Spindle rotation at 900 rpm
N40 ... X100 Z60              Tool tip positioned on diameter 100
N50 G92 S3000                 Spindle speed limited to 3000 rpm
N60 G96 S200                  Constant surface speed initialised
                              on X100
N70 ... X0                    Facing
N..
N..
N200 G97 S900                 Function G96 cancelled
N..
```

5.5.5 Spindle Range ► M40, M41, M42, M43, M44, M45

M40/M41/M42/M43/M44/M45 Spindle ranges.

The system allows the definition of six spindle ranges associated with address S.

Syntax

N.. [S..][M03/M04]**M40-M45**

S..	RPM spindle speed (with G97).
M03/M04	Spindle direction of rotation.
M40 to M45	Spindle range selection.

Properties of the Functions

Functions M40 to M45 are modal «pre» functions.

Cancellation

Functions M40 to M45 cancel one another.

Notes

The minimum and maximum speeds are defined for each range by the machine manufacturer.

Example:

- M40 = 50-500 rpm
- M41 = 400-900 rpm
- M42 = 800-4200 rpm

Example

```
N.. ...
N30 G97 S650 M41 M03                    Range M41
N..
```

Optionally, the spindle range can be determined automatically by programming the S address and the rpm.

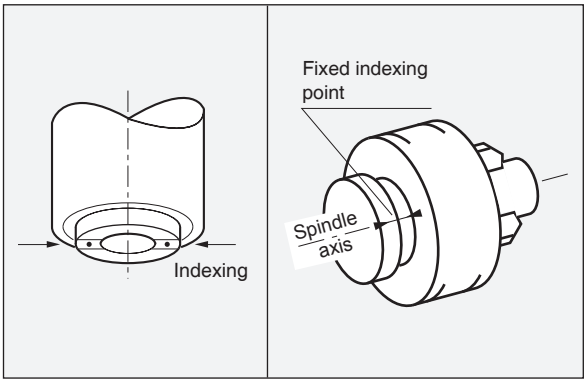
In case of ranges overlapping the selection Highest range or Lowest range is preset by the OEM (see machine operation manual).

Even with automatic range selection it is always possible to force a range as long as it is within the authorized values.

5.5.6 Indexed Spindle Stop ► M19

M19	Indexed spindle stop.

This function stops the spindle in a position defined with reference to a fixed point.



Syntax

N.. [S..] [M03/M04] [M40 to M45] EC±.. **M19**

S..	RPM spindle speed (with G97).
M03/M04	Spindle direction of rotation.
M40 to M45	Spindle speed ranges.
EC±..	Optional argument defining the indexing angle in degrees from the fixed point.
M19	Indexed spindle stop.

Properties of the Function

Function M19 is a modal «pre» function.

Cancellation

Function M19 is cancelled by one of functions M03, M04 or M05.

Notes

It is not necessary that the spindle be in rotation before requesting indexation. If the spindle is not rotating, indexing is carried out by positioning along the shortest path.

When the system includes bidirectional orientation capabilities, the stopped position is reached by the shortest path.

Example

Indexed spindle stop at + 90 degrees with reference to the origin.

```
N.. ...                    Tool call
N120 G97 S500 M03 M42      Spindle rotating
N130 EC90 M19              Indexed spindle stop
N..
```

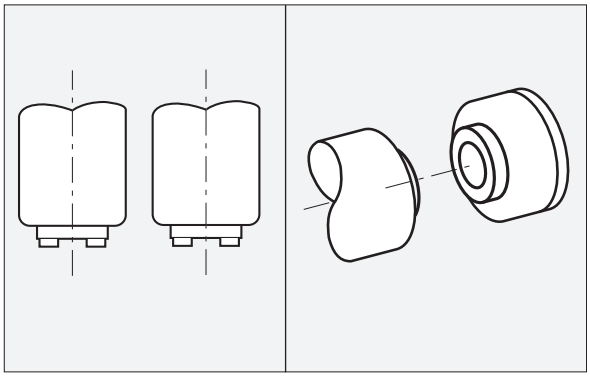
Turning Machine specific

Indexing the spindle on a lathe follows a similar programming except that the indexing position is introduced via address C instead of EC.

5.5.7 Spindle Control Selection ► M64, M65, M62, M63, M61

M64/M65/M62/M63
Control of spindles 1 to 4.
M61 release the current spindle.

When the machine is equipped with several spindles, these functions set the commands for the spindle servo-drives.



Syntax

N.. [S..] [M03/M04] [M40-M45] **M62-M65**

S..	Spindle speed in rpm (with G97).
M03/M04	Spindle direction of rotation.
M40-M45	Spindle speed ranges.
M64	Spindle 1 control selection.
M65	Spindle 2 control selection.
M62	Spindle 3 control selection.
M63	Spindle 4 control selection.

Properties of the Functions

Functions M62, M63, M64 and M65 are modal «pre» functions.

Spindle selection

Before selecting a spindle, it is mandatory that this spindle be released by the channel to which it belongs. Releasing a spindle is made by programming M61.

Cancellation

Functions M62, M63, M64 and M65 cancel one another.

At power on, after a reset or at the end of a program (M02), each spindle is assigned to its own channel (M64 is initialised if there is only one channel).

Notes

A spindle receives the functions of the channel to which it is assigned:

- Speed G97 S.., Constant surface speed if any (G96 S...)
- Speed limitation if any (G92 S...)
- Direction of rotation or spindle stop (M03, M04, M05),
- Spindle speed ranges (M40-M45),
- Spindle indexing (M19 EC..),
- Spindle speed override enabled or disabled (M48 or M49).

The spindle of a channel is released by:

- Selecting a new spindle (M62-M65),
- Release function M61.

When released from a channel, a spindle keeps all the characteristics it had when released but new functions for the channel are no longer addressed to it. They are addressed to the new spindle assigned to the channel if any.

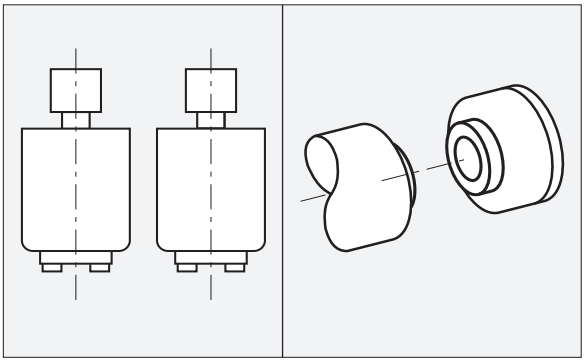
Example

```
N.. ...
N130 M65                          Assignment of spindle 2 to the channel
N140 G97 S500 M03 M40             Spindle 2 control selection
N..
```

5.5.8 Spindle Measurement Selection ► M66, M67, M68, M69

M66/M67/M68/M69
Spindle 1 to 4 measurement selection.

When the machine is equipped with several spindles, these functions are used for spindle measurement.



Syntax

N.. **M66/M67/M68/M69**

M66	Spindle 1 measurement.
M67	Spindle 2 measurement.
M68	Spindle 3 measurement.
M69	Spindle 4 measurement.

Properties of the Functions

Functions M66, M67, M68 and M69 are decoded modal «pre» functions.

Cancellation

Functions M66, M67, M68 and M69 cancel one another.

At power on, at the end of a program M02 or after a reset, the measurement of each spindle is assigned to its own channel (example: M66 is assigned to the channel 1). If there is no spindle with the same number as the channel, Spindle 1 is assigned by default (M66).

Notes

Each channel can use the measurement of any spindle.

Several channels can use the measurement of the same spindle.

If the measurement of a spindle is used by a channel for thread cutting, speed override by potentiometer for this spindle is inhibited during the thread cutting cycle (value forced to 100 percent).

When a declared spindle does not have an axis encoder, measurement of this spindle is simulated by the NC.

Example

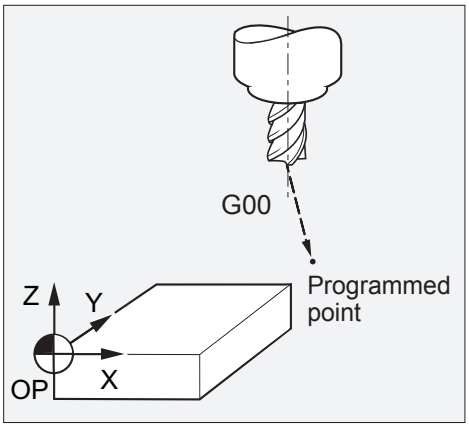
```
N.. ...
N180 M67                    Assignment of the spindle measurement
                            to channel 2
N190 G95 F0.15              Feed in mm/rev related to spindle 2
N..
```

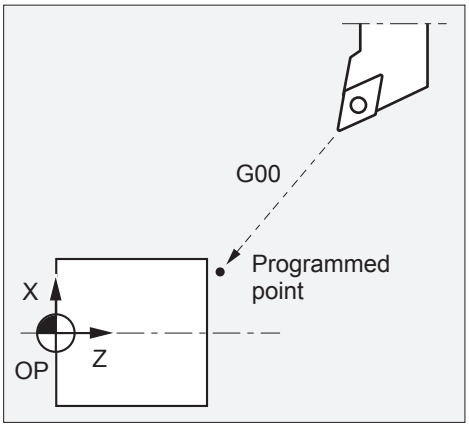
G00	Linear interpolation at high speed.

The programmed point is reached at high speed by a linear path.

The path is a combination of all the axis movements programmed in the block.

Programmable axes:
- X, Y, Z primary axes,
- U, V, W secondary axes,
- A, B, C rotary axes.





Syntax

N.. [G90/G91] **G00** [R-/R+] X.. Y.. Z..

G90/G91	Programming in absolute or incremental dimensions.
G00	Rapid positioning.
R-/R+	Defines a shift of the end point (R- before, R+ after) the value of the shift is equal to the radius of the current tool.
X.. Y.. Z..	End point: - Point coordinates with G90. - Value of the movement with G91.

Property of the Function

Function G00 is modal.

Cancellation

Function G00 is cancelled by one of functions G01, G02 or G03.

Notes

The speed along the path programmed in G00 is determined by the most solicited axis (it will run at its max speed).

Optional arguments R+ or R-:

- are active only in the block where they are programmed. The shift is applied in the interpolation plane.
- cannot be programmed in a block with PGP.

Programming of the additional axes and carrier/carried axis pairs

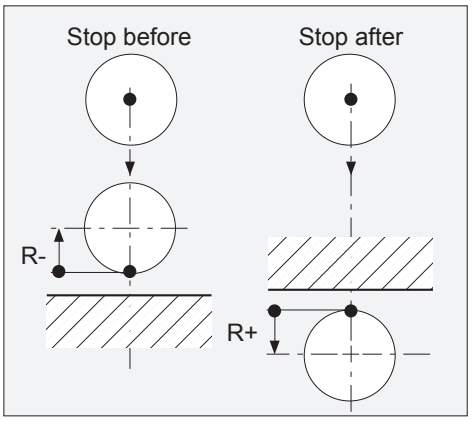
Two axes of carried/carrier axis pairs can be programmed in simultaneously G00 using function G52 (programming with reference to the machine origin).

Unlike in G1 programming a rotary axis at the current position will not result in a full turn.
The move is considered executed immediately (Positioning function).

5.6.1 Positioning with respect to the cutter dia. (R-/R+) example

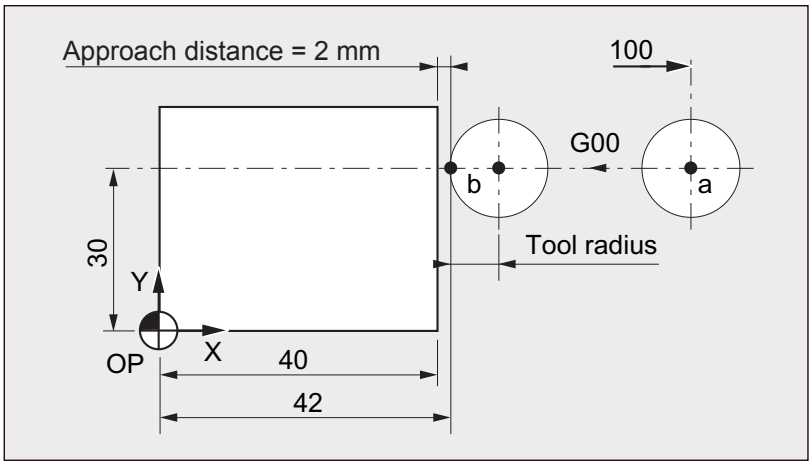
R+ or R- can be programmed to reach the specified approach point regardless of the read tool radius

Positioning is only applied to the axes of the plane



Example

Positioning with a 2mm guard

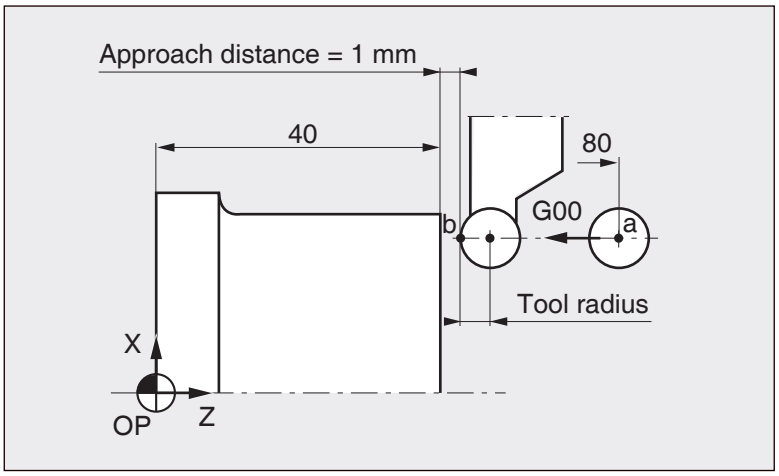



```
N.. ...
N.. D..
N140 G00 X100 Y30 Z-10
N150 R- X42
N160 ...
N..
```

Tool correction (see chapter 5.10)
Point a
Point b, stop before the programmed point

Example

Programming to stop at an approach distance of 1 mm




```
N.. ...
N.. D..                  Tool correction (see chapter 5.11)
N140 G00 X40 Z80         Point a
N150 R- Z41              Point b, stop before the programmed point
N160 ...
N..
```

5.7 Programming Movements

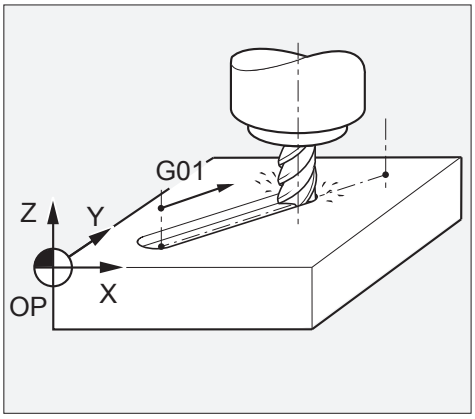
5.7.1 Linear Interpolation ► G01

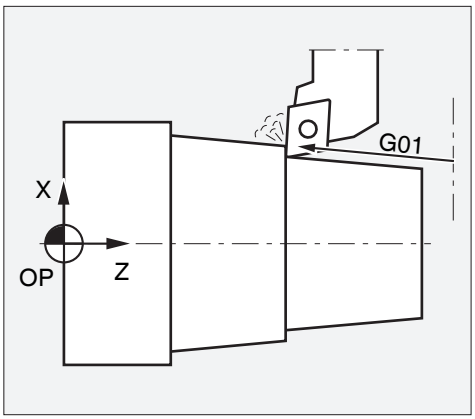
G01	Linear interpolation at the programmed feed rate.

The programmed point is reached by a linear path at the programmed feed rate.

The path is a combination of all the axis movements programmed in the block.

Programmable axes:
- X, Y, Z primary axes,
- U, V, W secondary axes,
- A, B, C rotary axes.





Syntax

N.. [G90/G91] **G01** [R+/R-] X.. Y.. Z.. [F..]

G90/G91	Programming in absolute or incremental dimensions.
G01	Linear interpolation at the programmed feed rate.
R-/R+	Shift end point before (R-) or after (R+) the programmed coordinate. The shift value is equal to the active tool radius. The distance is equal to the active tool radius.
X.. Y.. Z..	End point: - Point coordinates with G90. - Value of the movement with G91.
F..	Feed rate.

Properties of the Functions

Function G01 is modal. It is the default function.

Cancellation

Function G01 is cancelled by one of functions G00, G02 or G03.

Notes

Optional arguments R+ or R-:
- are active only in the block where they are programmed,
- cannot be programmed in a block with PGP.

5.7.1.1 Programming of additional axes and carrier/carried axis pairs

Linear interpolations can be executed as combined movements on primary and additional axes.

Example:

Linear interpolation on a primary axis and a secondary axis.

N.. G01 Y.. W.. F..

Linear interpolation on a secondary axis and a rotary axis.

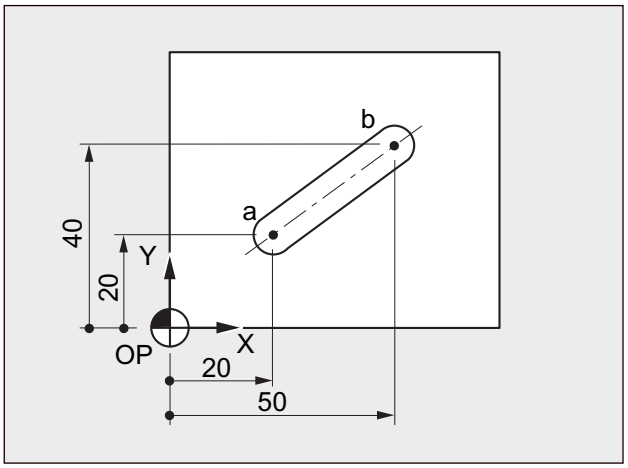
N.. G01 U.. C.. F..

Carrier / carried axis can only be programmed simultaneously in G00 and G52. Any attempt in G01 will result in error 139..

Programming a rotary axis at its current position in G01 will result in a 360° movement, unless parameter E11008 is set to 0.

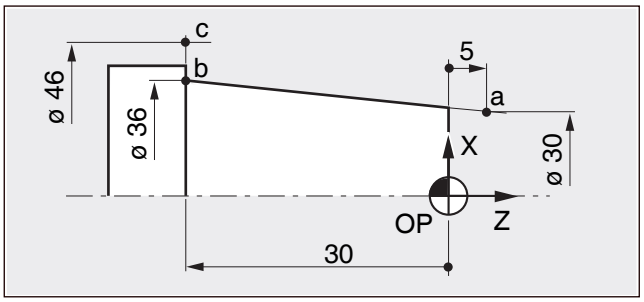
Example

Programming in absolute dimensions (G90)




```
20
N10 G00 G52 Z..          Rapid to tool change position
N20 ... D8               Tool call offset selection
N30 S600 M40 M03
N40 X20 Y20 Z2           Approach point a
N50 G01 Z-1.5 F50        Penetration in Z
N60 X50 Y40 F120         Point b
N..
```

Turning and facing in absolute programming (G90)

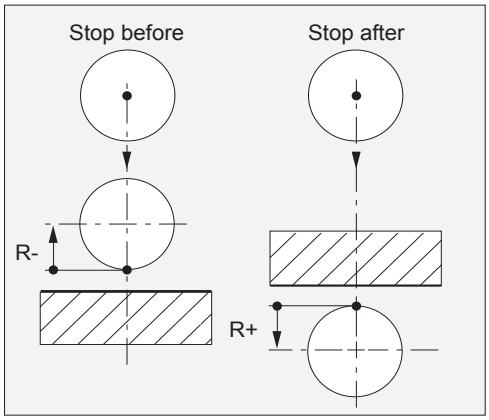



```
%20
N10 G00 G52 X.. Z..          Tool change position
N20 ...                      Tool call
N30 G97 S600 M40 M04
N40 X30 Z5                   Point a, approach
N50 G96 S200
N60 G95 F0.2                 Feed rate in mm/rev.
N70 G01 X36 Z-30             Point b, turning
N80 X46 F0.1                 Point c, facing
N..
```

5.7.2 Positioning with respect to the cutter dia. (R+/R-)

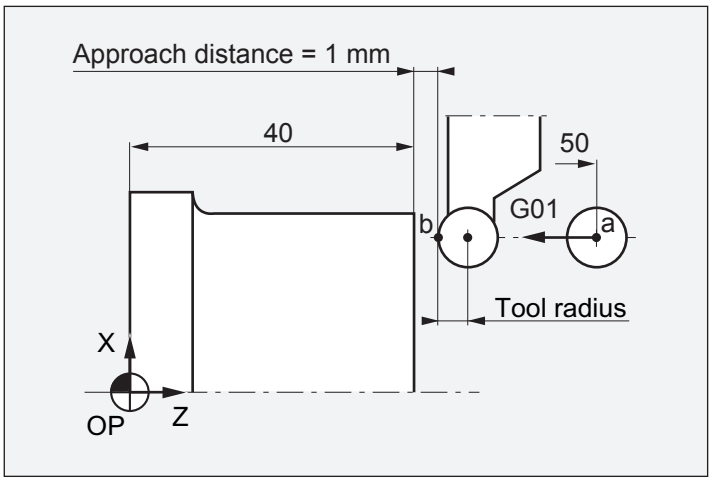
R+ or R- can be programmed to reach the specified approach point regardless of the read tool radius.

Positioning is only applied to the axes of the active plane.



Example

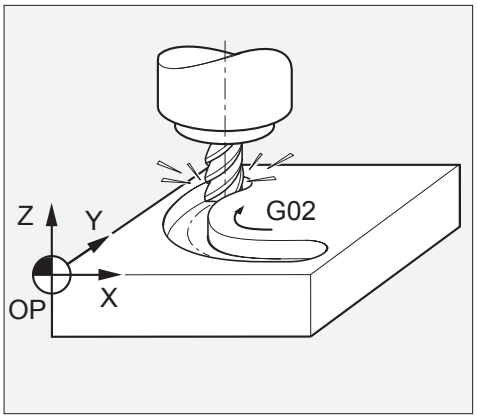
Programming to stop at an approach distance of 1 mm

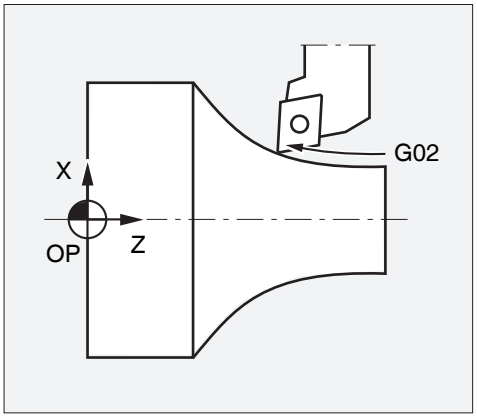



```
N.. ...
N.. D..                    Tool correction (see chapter 5.11)
N140 G00 X40 Z50           Point a
N150 G01 R- Z41 G95 F0.2   Point b, stop before the programmed point
N..
```

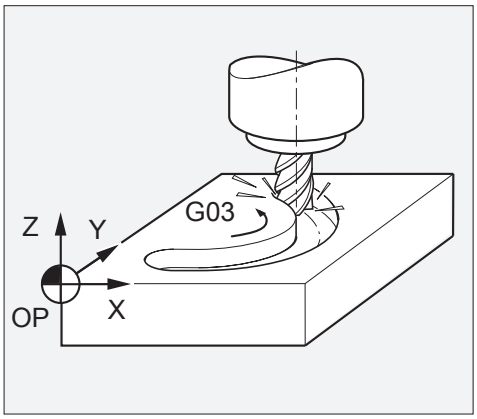
5.7.3 Circular Interpolation ► G02, G03

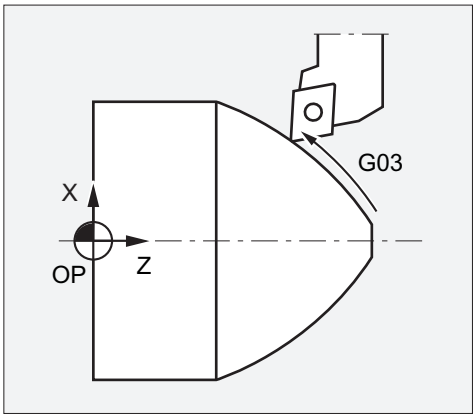
G02	Clockwise circular interpolation at the programmed feed rate.





G03	Counterclockwise circular interpolation at the programmed feed rate.





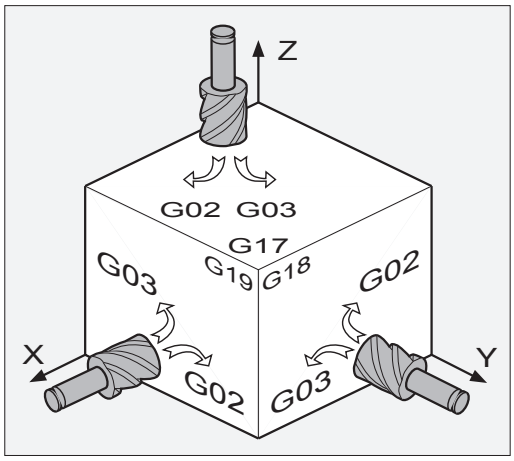
- The programmed point is reached by a circular path.
- The programmed feed rate is applied at the tool center.
- The two controlled linear axes depend on the choice of the interpolation plane.

For Milling Machine

- X (or U) and Y (or V) axes for G17
- Z (or W) and X (or U) axes for G18
- Y (or V) and Z (or W) axes for G19.

For Turning Machine

- Z (or W) and X (or U) in G20
- X and Y in G21
- Y and Z in G22.



Syntax (XY plane)

N.. [G17] [G90/G91] **G02/G03** X.. Y.. I..J.. / R.. [F..]

G17	XY plane selection
G90/G91	Programming in absolute or incremental dimensions
G02	Clockwise circular interpolation
G03	Counterclockwise circular interpolation
X.. Y..	End point. - Coordinates of the end point with G90. - Value of the movement with G91.
I.. J..	Location of the circle centre in the XY plane (I along X, J along Y). - With reference to the program origin with G90. - With reference to the interpolation start point with G91.
R..	Radius of the circle.
F..	Feed rate (see chapter 5.10).

Syntax according to the plane selected: G17/G18/G19:

Main interpolation plane	Function	Interpolation	Syntax
XY	G17	G02/G03	X.. Y.. I.. J.. or R..
ZX	G18	G02/G03	X.. Z.. I.. K.. or R..
YZ	G19	G02/G03	Y.. Z.. J.. K.. or R..

Properties of the Functions

Functions G02 and G03 are modal.

Cancellation

Function G02 is cancelled by functions G00, G01 or G03.

Function G03 is cancelled by functions G00, G01 or G02.

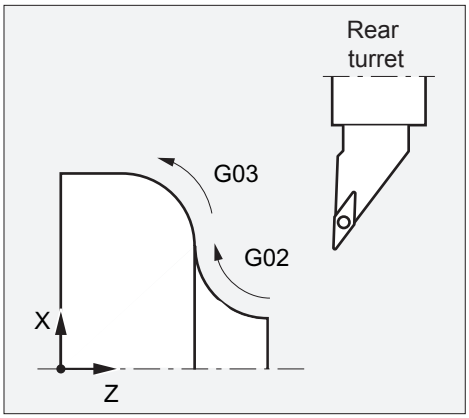
Notes

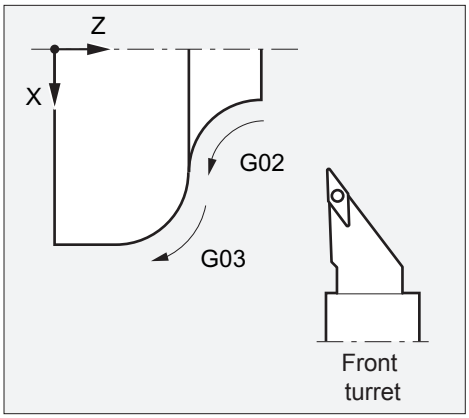
The third axis in the plane if any can be programmed with circular interpolation to give helical interpolation.

In a block programmed in G02 or G03, all the addresses used for the interpolation must be specified even if they are zero (I0, J0; Plane G17) or unchanged with respect to the previous block (X and Y; Plane G17).

5.7.3.1 Direction of movement according to the turret position with Turning Machine

Lathe with one turret

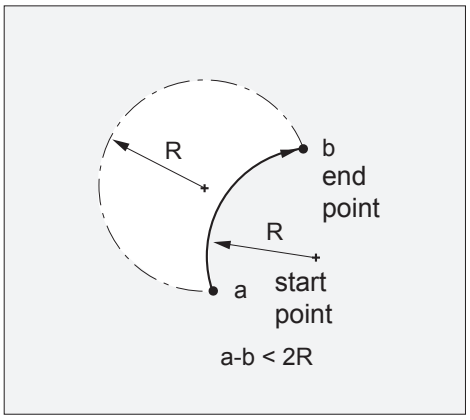




5.7.3.2 Programming a circle by its radius (R..)

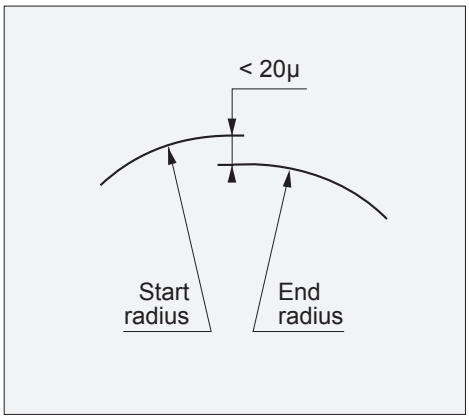
The system chooses the path whose angle is less than 180 degrees (a path with an angle greater than 180 degrees can only be obtained by programming the circle by the coordinates of its centre or in PGP).

If the distance between the start point and end point is greater than twice the programmed radius, the system generates an error message.



5.7.3.3 Programming a circle by the coordinates of its centre (I and J, plane G17)

When the difference in radii between the start point and end point is greater than 20 µ, the system generates an error message.



5.7.3.4 Programming the additional axes and carrier/carried axis pairs

Circular interpolation can be carried out in the plane selected on the following primary and secondary axis pairs:

Function	Main interpolation plane	Axis pairs
G17	XY	XY, UV, UY, XV
G18	ZX	ZX, WU, WX, ZU
G19	YZ	YZ, VW, YW, VZ

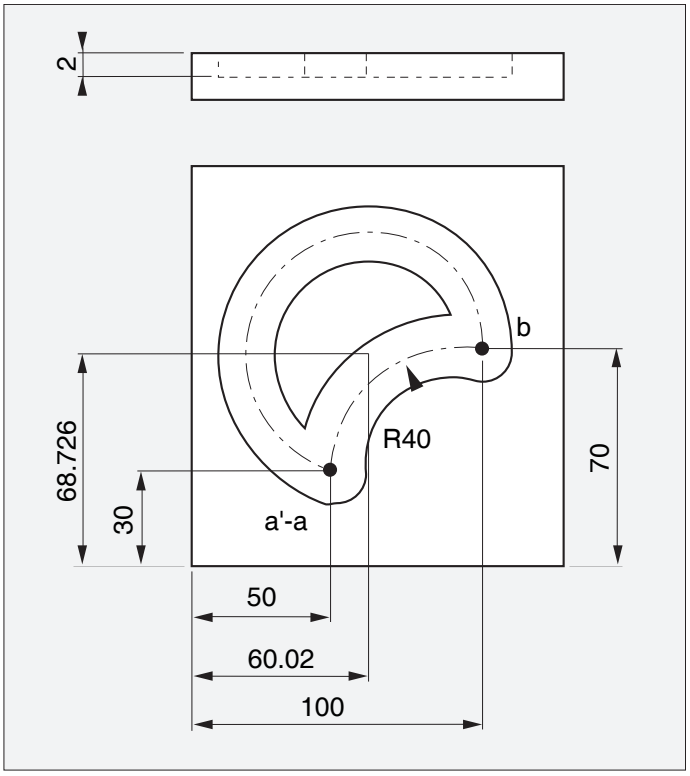
Other axes such as rotary and/or linear axes can be associated in a circular interpolation block, but these axes are interpolated linearly.

A movement on a carrier axis causes an identical movement on the carried axis.

Example

Circular interpolation using two methods

Circular interpolation G02 is carried out by programming the radius (R) and circular interpolation G03 is carried out by programming the circle centre (with I and J).

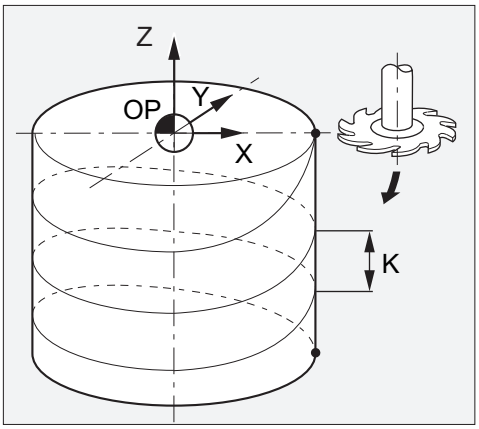



```
%23
N10 G00 G52 Z..                          Rapid to tool change
N20 ...                                  Tool call
N30 S300 M40 M03
N40 X50 Y30 Z2                           Approach point a
N50 G01 Z-2 F50                          Penetration in Z
N60 G02 X100 Y70 R40 F150                Point b
N70 G03 X50 Y30 I60.02 J68.726           Point a'
N80 G00 G52 Z.. M05
N90 M02
```

5.7.4 Helical Interpolation

Helical interpolation is a combination of circular and linear interpolation. The tool axis describes a helix with a constant pitch.

Helical interpolation can be carried out in all three planes and applies to the primary and secondary axes.



Syntax (XY plane)

N.. [G17] [G90/G91] **G02/G03** X.. Y.. Z.. I.. J.. / R.. K.. [F..]

G17	G17 XY plane selection.
G90/G91	Programming in absolute or incremental dimensions.
G02	Clockwise helical interpolation.
G03	Counterclockwise helical interpolation.
X.. Y..	End point in the XY plane. End point coordinates with G90. Value of the movement with G91.
Z..	Point to be reached on the helix axis (XY plane) with G90 or G91.
I.. J..	Position of the interpolation centre in the XY plane (I along X, J along Y), with reference to the program origin with G90 or the interpolation start point with G91.
R..	Radius of the circle to be interpolated.
K..	Approximate helix pitch in Z (unsigned value).
F..	Feed rate in mm/min or inches/min.

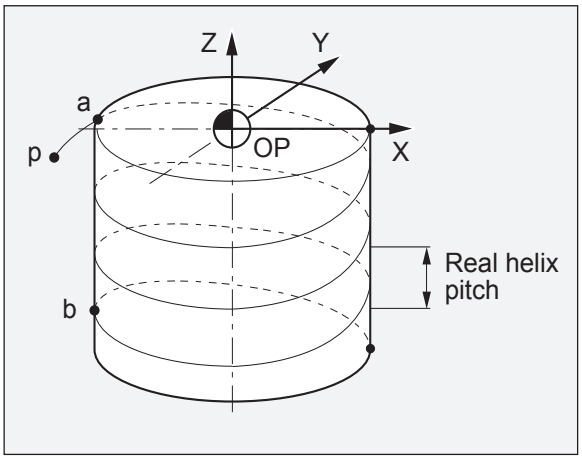
Notes

With helical interpolation:

- The helix pitch applies only to the axis normal to the interpolation plane of the basic reference system.
- Programming of the pitch allows the system to determine the number of revolutions required so that the real pitch is as close as possible to the programmed pitch considering the position of the start and end points.

Example

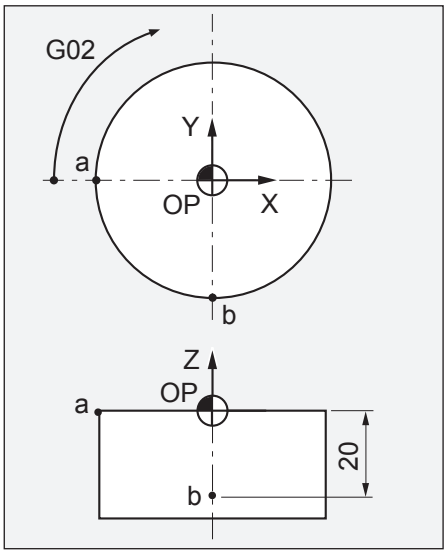
Helical interpolation in the XY plane with programming in absolute dimensions (G90)




```
%55
N10 G00 G52 Z..                    Rapid to tool change
N20 ...                            Tool call
N30 S200 M40 M03
N40 G00 Xp Yp Zp                   Approach point p
N50 G01 Xa Ya Za                   Start point a
N60 G02 Xb Yb Zb I0 J0 K.. F150    End point
N.. G00 X..
N..
```

5.7.5 Determination of the helix depth

Helix depth in G02 with different helix pitches:
Assume difference between a and b in Z: Δ Z = 20 mm.
Δ Z / K = Approximate number of revolutions.



The final X,Y, Z position is always accurate. If the programmed pitch does not match with the axial displacement, the system will recalculate the closest value. No error will be generated.

5.7.5.1 Syntax according to the plane selected: G17/G18/G19

Interpolation plane	Interpolation	Syntax	Pitch
G17(XY)	G02/G03	X.. Y.. I.. J.. or R..	K
G18 (ZX)	G02/G03	Z.. X.. K.. I.. or R..	J
G19 (YZ)	G02/G03	Y.. Z.. J.. K.. or R..	I

5.7.5.2 Programming of additional axes and carrier/carried axis pairs

Other axes such as rotary and/or linear axes can be associated in a block, but these axes are interpolated linearly. Carrier/carried axes cannot be programmed simultaneously.

Example

```
N..
N60 G17
N70 G02 X.. V.. W.. A.. I.. J.. K..
N..
```

- X and V linear axes interpolated circularly
- Linear axis W and rotary axis A interpolated linearly
- Helix pitch K applied to the W axis.

Example

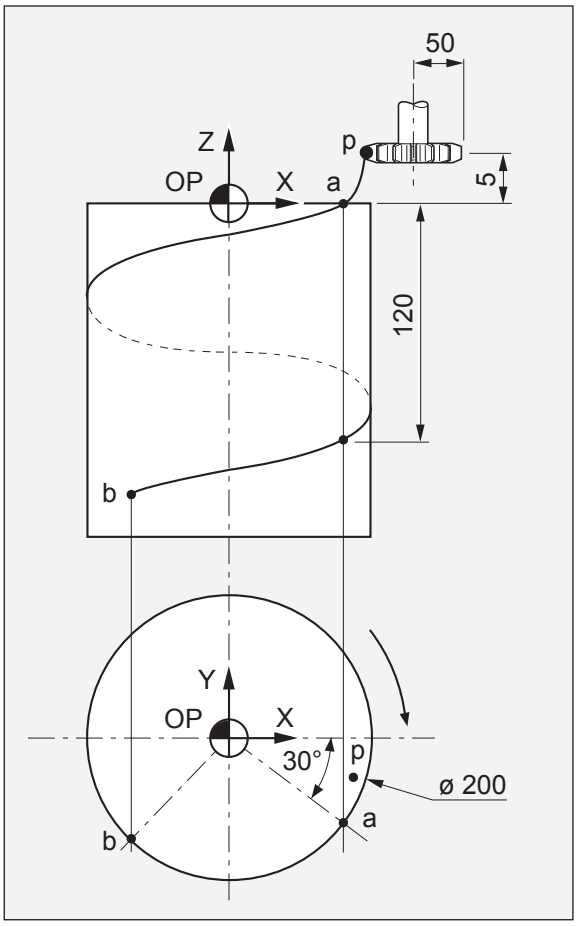
```
N..
N80 G18
N90 G03 X.. Y.. U.. V.. W.. I.. J..
N..
```

- X and W linear axes interpolated circularly
- Y, U, V linear axes interpolated linearly
- Helix pitch J applied to the Y axis.

This type of programming is only possible when the X, U, Y, V axes are independent.

Example

Helical interpolation in the XY plane



Execution of a helix on a cylindrical part with a diameter of 200 mm (radius = 100):

- Helix depth = 2 mm
- Real helix pitch = 120 mm
- Positioning clearance in point p on Z = 5 mm
- Start point a: Position Z = 0, angular position = -30 degrees
- End point b: Position Z = -155

Angular distance from a to b: 360° x 155 / 120 = 465° (i.e. one revolution and 105°)

Distance from end point b to point a: -105° + (-30°) = -135°

Position of point p: 360° x 5 / 120 = 15° i.e. p = 30° + 15° = -15°

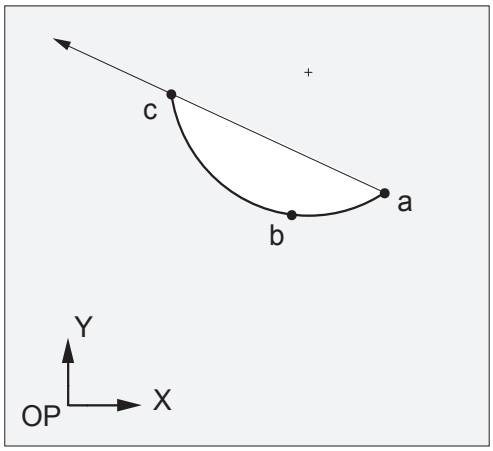
```
%58
N10 G00 G52 Z..                          Rapid to tool change
N20 ...                                  Tool call
N30 S200 M40 M03
N40 G00 X110 Y110 Z5                     Approach position
N50 G01 G41 X94.661 Y25.364              Point p, (start) in G41 (see chapter 5.11)
N60 G02 X-69.296 Y-69.296 Z-155 I0 J0    Point b, (end)
K120 F300
N70 G01 G40 X-110 Y-110
N..
```

5.7.6 Circular Interpolation Defined by Three Points ► G23

G23	Circular interpolation defined by three points.

Circular interpolation can be carried out by programming:

- the start point (defined in the block preceding function G23),
- the end point and the intermediate point (defined in the same block as function G23).



The direction of circular interpolation is defined by the position of the intermediate point (b) with respect to the start point (a) and the end point (c), i.e.:

- to the left of line ac: clockwise,
- to the right of line ac: counterclockwise.

Syntax (XY plane)

N.. [G90/G91] **G23** X.. Y.. I.. J.. [F..]

G90/G91	Absolute or incremental programming.
G23	Clockwise or counterclockwise circular interpolation.
X.. Y..	End point.
I.. J..	Intermediate point.
F..	Feed rate.

Properties of the Function

Function G23 is nonmodal.

Cancellation

Function G23 is cancelled at the end of the block.

Notes

The arguments of function G23 must not be separated by any other address. Otherwise, the system returns error message 101.

G23 is not available in G21 status (cartesian to polar interpolation).

Example:

```
N.. G23 X.. Y.. F.. I.. J..
```
Programming incorrect

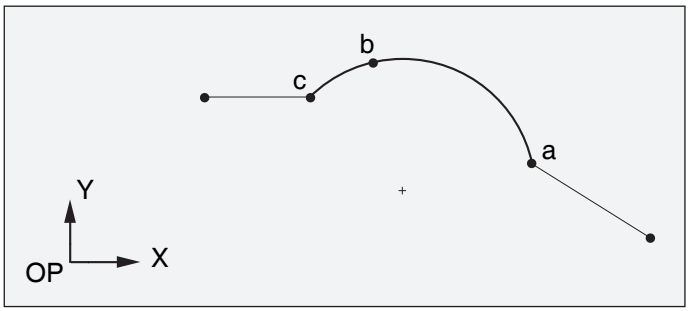
Circular interpolation defined by G23 can be in absolute dimensions (G90) or incremental dimensions (G91).

5.7.6.1 Syntax According to the Plane Selected: G17/G18/G19

Main interpo lation plane	Function	Intermediate point	Syntax
XY	G17	I J	G23 X.. Y.. I.. J..
ZX	G18	K I	G23 Z.. X.. K.. I..
YZ	G19	J K	G23 Y.. Z.. J.. K..

Example

From A to C through B in G17




```
N..
N130 Z..                          Position on Z
N140 G01 Xa Ya F150               Point a, approach
N150 G23 Xc Yc Ib Jb F100         Circular interpolation
N160 G01 X.. Y.. F150
N..
```

5.7.7 Polar Programming

Polar programming is used to define paths or positions when one of the part dimensions is angular.

General

Polar programming is used for line or circle geometric elements defined in the same plane.

Polar programming:
- can coexist with ISO programming and Profil Geometry Programming (PGP)
- can be in absolute dimensions (G90), incremental dimensions (G91) or a combination thereof (G90 and G91, possibly in the same block)
- can be combined with cartesian programming.

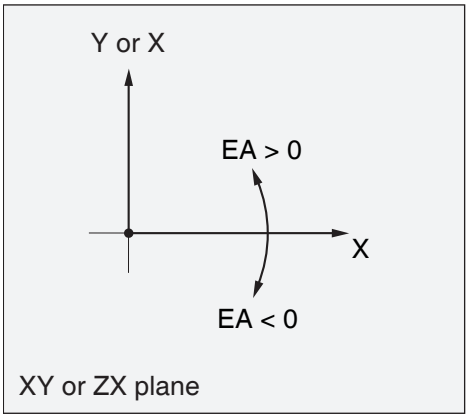
Polar programming can be carried out in one of the interpolation planes selected: XY, ZX or YZ (in all cases, the line or circle start point must be defined in the same plane as the one used in programming).

Reference Axis of the Polar Angle

For polar programming of a line or circle, the polar angle is always programmed by argument EA (whatever the interpolation plane selected).

The polar angle is defined with respect to the reference axis of the interpolation plane:
- X in the XY plane (G17)
- Z in the ZX plane (G18)
- Y in the YZ plane (G19).

The positive or negative direction of the angle is defined in the trigonometric circle.

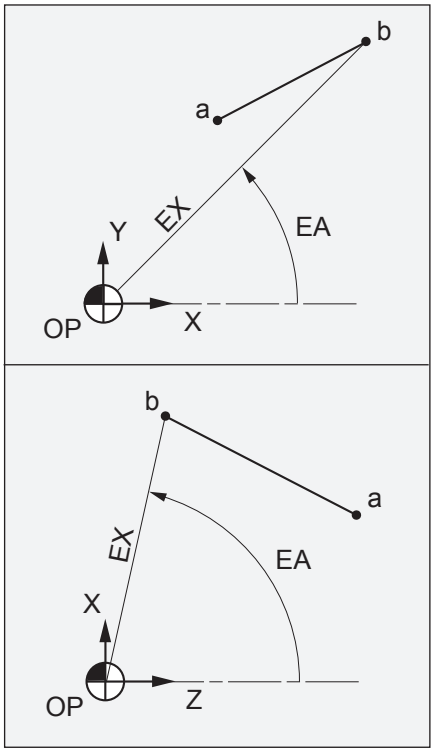


5.7.7.1 Polar Programming of a Line

A line can be programmed:
- in absolute dimensions with function G90
- in incremental dimensions with function G91.

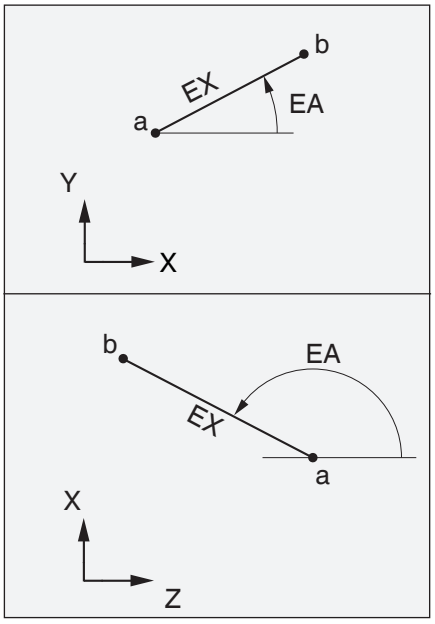
Polar Programming of a Line in Absolute Dimensions (G90)

The line is defined in absolute dimensions by:
- its start point (a) contained in the block preceding the polar programming block
- the polar coordinates of its end point (b) defined with respect to the program origin (OP).



Polar Programming of a Line in Incremental Dimensions (G91)

The line is defined in incremental dimensions by:
- its start point (a) contained in the block preceding the polar programming block
- the polar coordinates of its end point (b) defined incrementally with respect to its start point (or the last programmed point).



Syntax (XY plane)

N.. [G17] [G90/G91] G00/G01 EA.. EX.. [F..]

G17	XY plane selected.
G90/G91	Programming in absolute or incremental dimensions.
G00/G01	Linear interpolation.
EA..	Angle of line EX.
EX..	Length of the line. In G90: EX = distance from program origin to end point. In G91: EX = distance from start point to end point.
F..	Feed rate (see chapter 5.10).

Notes

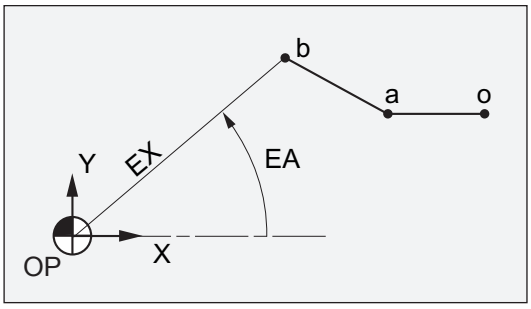
In the block, EA must be programmed first, before EX.

Argument EX:
- is always addressed by the same letters (i.e. EX), whatever the interpolation plane
- must always be programmed as a positive value.

Example

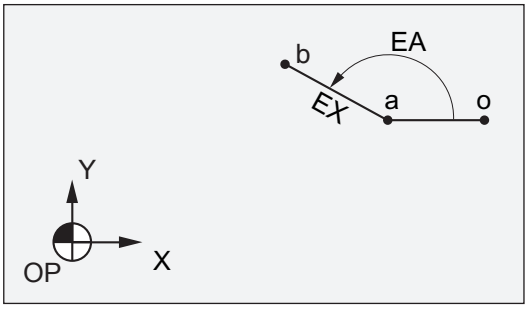
Line programmed in absolute dimensions

```
N..
N.. G90 G17
N200 X60 Y10 (Point o)
N210 G01 X40 Y10
N220 EA30 EX35
N..
```



Line programmed in incremental dimensions example

```
N..
N.. G90 G17
N120 X60 Y10 (Point o)
N130 G01 X40 Y10
N130 G91 EA120 EX15
N..
```

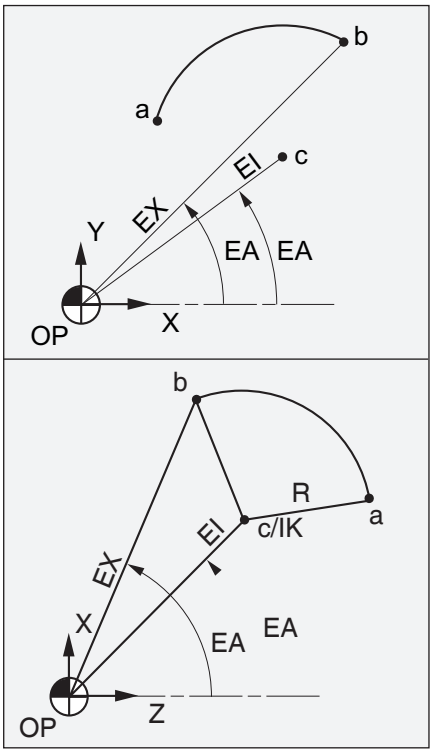


5.7.7.2 Polar Programming of a Circle

A circle can be programmed:
- in absolute dimensions with function G90 or incremental dimensions with function G91,
- as a combination of cartesian and polar coordinates,
- as a combination of incremental and absolute dimensions (G90/G91) and cartesian and polar coordinates,
- by its arc angle and its centre defined in cartesian or polar coordinates.

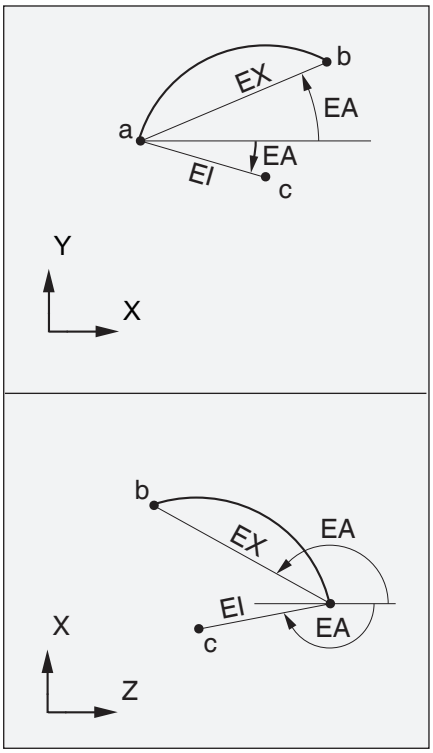
Polar Programming of a Circle in Absolute Dimensions (G90)

The circle is defined in absolute dimensions by:
- its start point (a) contained in the block preceding the polar programming block,
- the polar coordinates of the end point (b) and the centre (c) defined in absolute dimensions with respect to the program origin (OP).



Polar Programming of a Circle in Incremental Dimensions (G91)

The circle is defined in incremental dimensions by:
- its start point (a) contained in the block preceding the polar programming block,
- the polar coordinates of the end point (b) and the centre (c) defined in incremental dimensions with respect to the circle start point (or last point programmed).



Syntax (XY plane)

N.. [G17] [G90/G91] G02/G03 EA.. EX.. EA.. EI.. [F..]

G17	XY plane selected.
G90/G91	Programming in absolute or incremental dimensions.
G02/G03	Circular interpolation.
EA..	Angle of line EX.
EX..	Length of the line. In G90 : EX = distance from program origin to end point. In G91: EX = distance from start point to end point.
EA..	Angle of line EI.
EI..	Length of the line. In G90 : EI = distance from program origin to circle centre. In G91 : EI = distance from start point to circle center.
F..	Feed rate (see chapter 5.10).

Notes

The following notes apply to all cases of circles programmed in absolute or incremental dimensions.

In a block, the programming order must be:
- End point EA then EX
- Centre point EA then EI.

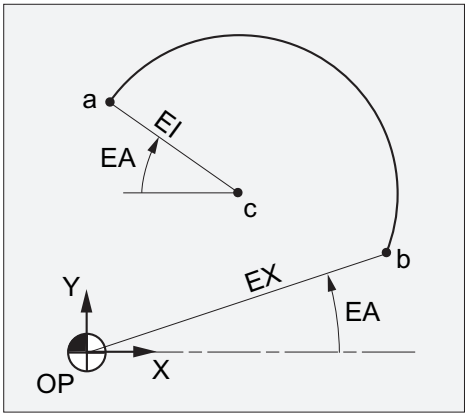
Arguments EX and EI must always be programmed in the positive direction.

Arguments EX and EI are addressed by these letters whatever the interpolation plane chosen.

Examples

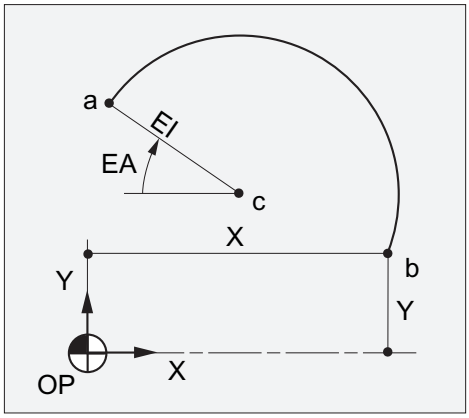
End point in Polar Absolute Center in Polar incremental

```
N.. G90 G17 G01 Xa Ya
N.. G02 EAb EXb G91 EAc EIc
```



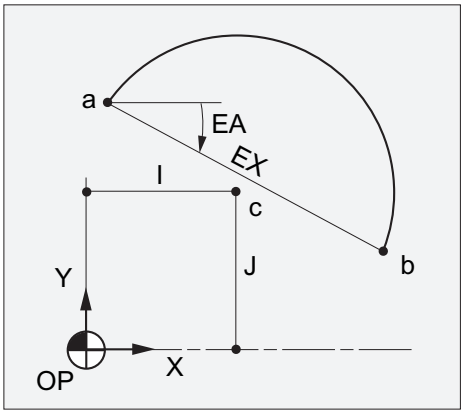
End point in Cartesian Absolute Center in Polar incremental

```
N.. G90 G17 G01 Xa Ya
N.. G02 Xb Yb G91 EAc EIc
```



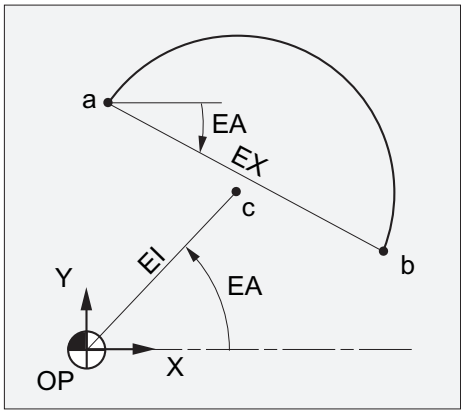
End point in Polar Incremental Center in Cartesian Absolute

```
N.. G90 G17 G01 Xa Ya
N.. G91 G02 EAb EXb G90 Ic Jc
```



End point Polar Incremental Center in Polar Absolute

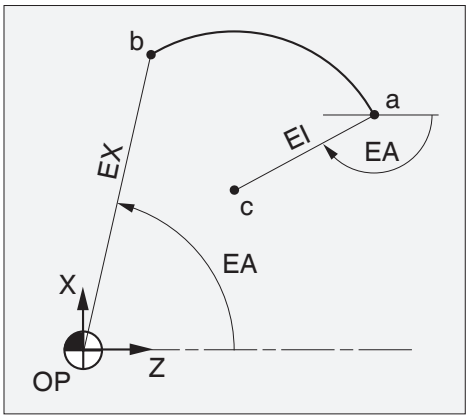
```
N.. G90 G17 G01 Xa Ya
N.. G91 G02 EAb EXb G90 EAc EIc
```



Examples

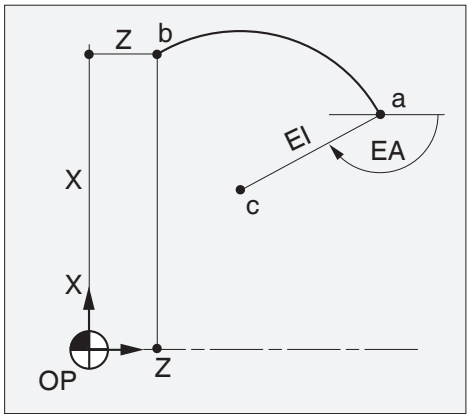
End point in Polar Absolute Center in Polar incremental

```
N.. G90 G20 G01 Xa Za
N.. G03 EAb EXb G91 EAc EIc
```



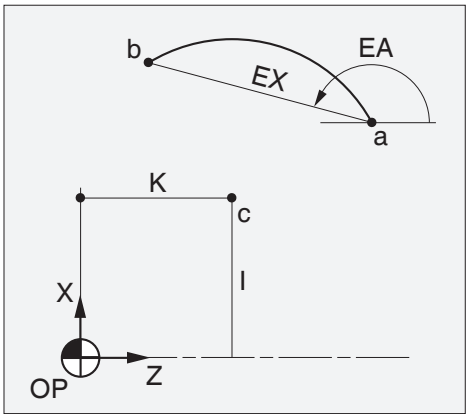
End point in Cartesian Absolute Center in Polar incremental

```
N.. G90 G20 G01 Xa Za
N.. G03 Xb Zb G91 EAc EIc
```



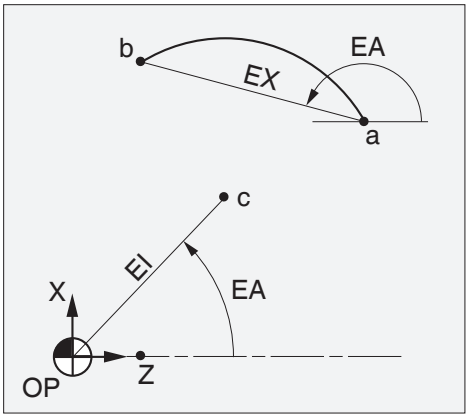
End point in Polar Incremental Center in Cartesian Absolute

```
N.. G90 G20 G01 Xa Za
N.. G91 G03 EAb EXb G90 Ic Kc
```



End point Polar Incremental Center in Polar Absolute

```
N.. G90 G20 G01 Xa Za
N.. G91 G03 EAb EXb G90 EAc EIc
```



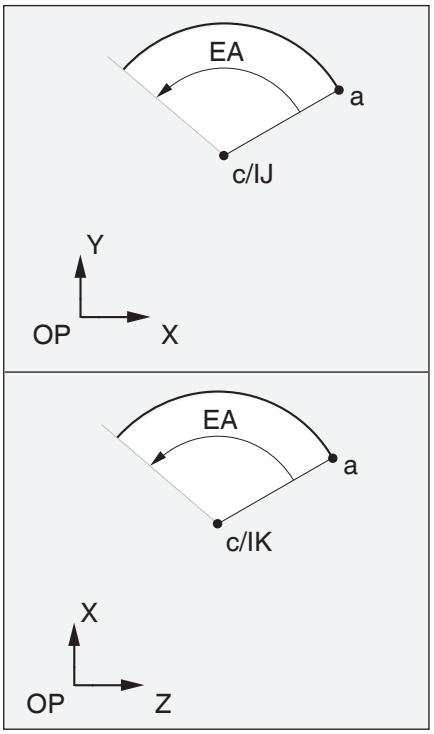
5.7.7.3 Defining a Circle by the Arc Angle

Defining a circle by the arc angle and cartesian programming of its centre defined in absolute or incremental dimensions

The circle is defined by:
- its start point (a) contained in the block preceding the arc angle programming block,
- the cartesian coordinates of its centre (c) and its arc angle.

The centre can be programmed in:
- absolute dimensions with G90,
- incremental dimensions with G91.

The arc angle EA is defined in absolute dimensions.



Syntax (XY plane)

N.. [G17] [G90/G91] G02/G03 I.. J.. EA.. [F..]

G17	XY plane selection.
G90/G91	Programming of the circle centre in absolute or incremental dimensions.
G02/G03	Circular interpolation.
I.. J..	Cartesian coordinates of the circle centre in the XY plane (I along X and J along Y): - in G90 with reference to the program origin, - in G91 with reference to the circle start point.
EA..	Arc angle. Angle of the end point with respect to the start point.
F..	Feed rate.

Notes

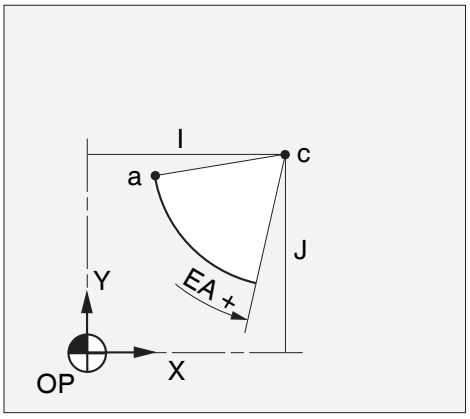
The notes below concern only circles defined by the arc angle (with the centre defined in cartesian coordinates).

When a value of zero is assigned to EA, the system describes a complete circle.
When the circle is defined from X to Y, EA is positive; in the opposite direction, EA is negative.

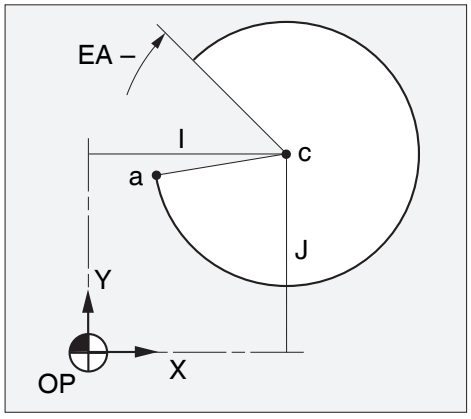
Four types of circles are possible in absolute programming, depending on the direction of the programmed circular interpolation (G02 or G03) and the sign (positive or negative) of arc angle EA.

Examples

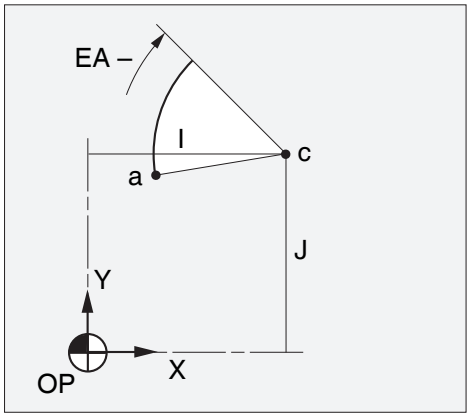
```
N.. G90 G17 Xa Ya
N.. G03 I.. J.. EA+75
```



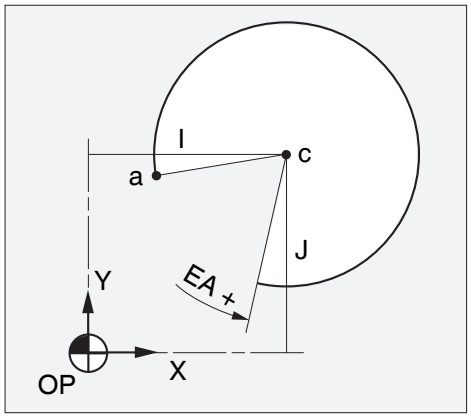

```
N.. G90 G17 Xa Ya
N.. G03 I.. J.. EA-75 (or EA+285)
```




```
N.. G90 G17 Xa Ya
N.. G02 I.. J.. EA-75
```




```
N.. G90 G17 Xa Ya
N.. G02 I.. J.. EA+75 (or EA-285)
```



Syntax (ZY plane)

N.. [G20] [G90/G91] G02/G03 I.. K.. EA.. [F..]

G20	ZX plane selection.
G90/G91	Programming of the circle centre in absolute or incremental dimensions.
G02/G03	Circular interpolation.
I.. K..	Coordinates of circle centre in the ZX plane (I along X and K along Z): - in G90 with reference to the program origin - in G91 with reference to the circle start point.
EA..	Arc angle. Angle of the end point with respect to the start point.
F..	Feed rate.

Notes

The notes below concern only circles defined by the arc angle (with the centre defined in cartesian coordinates).

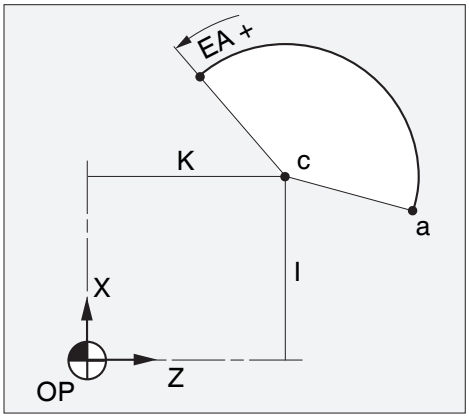
When a value of zero is assigned to EA, the system describes a complete circle.

When the circle is defined from Z to X, EA is positive; in the opposite direction, EA is negative.

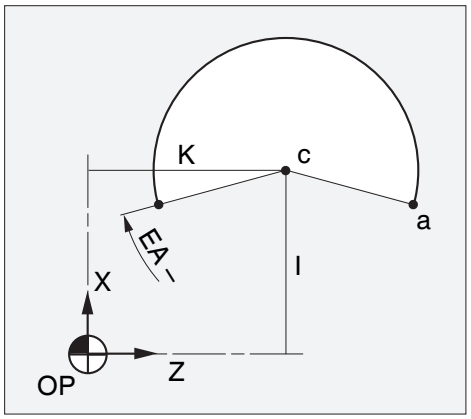
Four types of circles are possible in absolute programming, depending on the direction of the programmed circular interpolation (G02 or G03) and the sign (positive or negative) of arc angle EA.

Examples

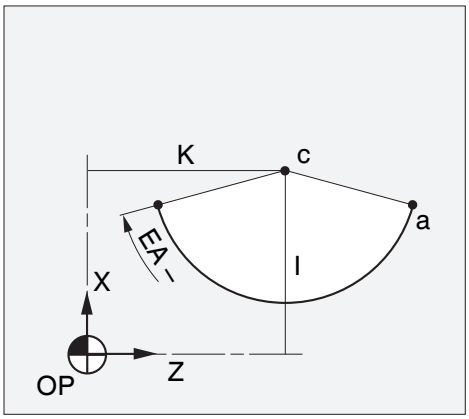
```
N.. G90 G20 Xa Za
N.. G03 I.. K.. EA+160
```



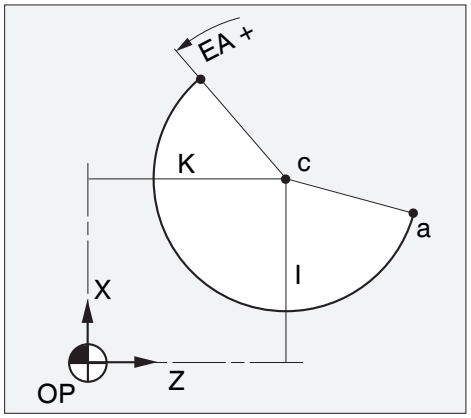

```
N.. G90 G20 Xa Za
N.. G03 I.. K.. EA-160 (ou EA+200)
```




```
N.. G90 G20 Xa Za
N.. G02 I.. K.. EA-160
```



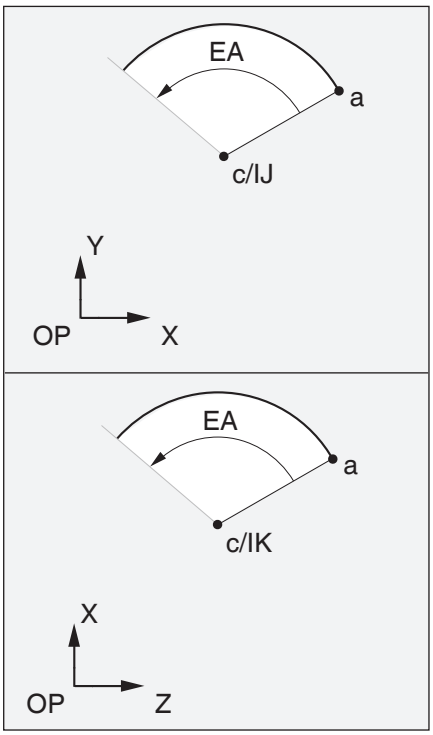

```
N.. G90 G20 Xa Za
N.. G02 I.. K.. EA+160 (ou EA-200)
```



Defining a circle by its arc angle and polar programming of its centre in absolute coordinates (G90)

The circle is defined by:
- its start point (a) contained in the block preceding the arc angle programming block
- the polar coordinates of its centre (c) and its arc angle.

Arc angle EA is defined in absolute dimensions.



Syntax (XY plane)

N.. [G17] [G90] G02/G03 EA.. EI.. EA.. [F..]	
G17	XY plane selection.
G90	Circle centre programmed in absolute dimensions.
G02/G03	Circular interpolation.
EA..	Angle of line EI.
EI..	Length of the line. EI = distance from program origin to circle centre.
EA..	Arc angle. Angle of the end point with respect to the start point.
F..	Feed rate.

Notes

The notes below concern only circles defined by the arc angle (with the centre defined in polar coordinates).

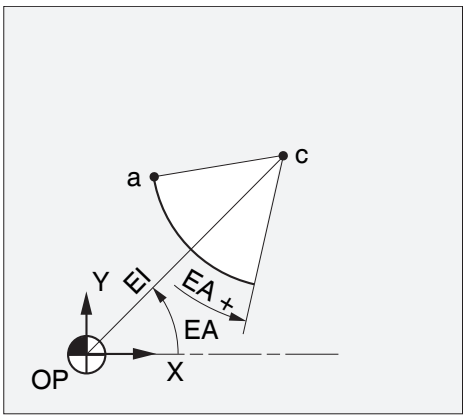
When a value of zero is assigned to EA, the system describes a complete circle.

When the circle is defined from X to Y, EA is positive; in the opposite direction, EA is negative.

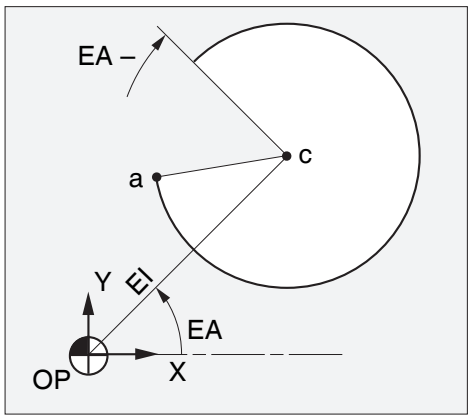
Four types of circles are possible in absolute programming, depending on the direction of the programmed circular interpolation (G02 or G03) and the sign (positive or negative) of arc angle EA.

Example

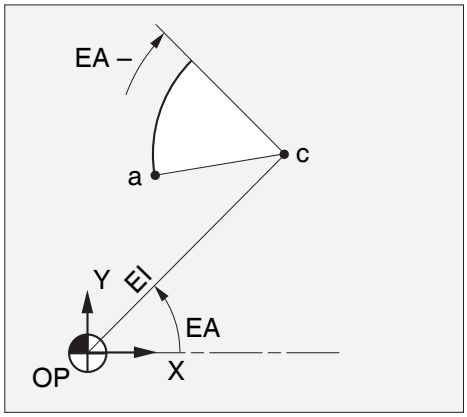
```
N.. G90 G17 Xa Ya
N.. G03 EA.. EI.. EA+70
```



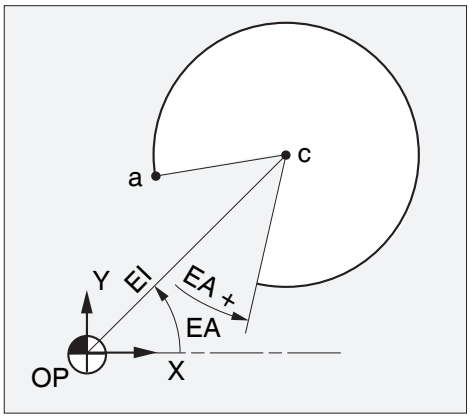

```
N.. G90 G17 Xa Ya
N.. G03 EA.. EI.. EA-70 (or EA+290)
```




```
N.. G90 G17 Xa Ya
N.. G02 EA.. EI.. EA-70
```




```
N.. G90 G17 Xa Ya
N.. G02 EA.. EI.. EA+70 (or EA-290)
```



Syntax

N.. [G20] [G90] G02/G03 EA.. EI.. EA.. [F..]	
G20	ZX plane selection.
G90	Circle centre programmed in absolute dimensions.
G02/G03	Circular interpolation.
EA..	Angle of line EI. program origin/circle centre
EI..	Length of the line. EI = distance from program origin to circle centre.
EA..	Arc angle. Angle of the end point with respect to the start point.
F..	Feed rate.

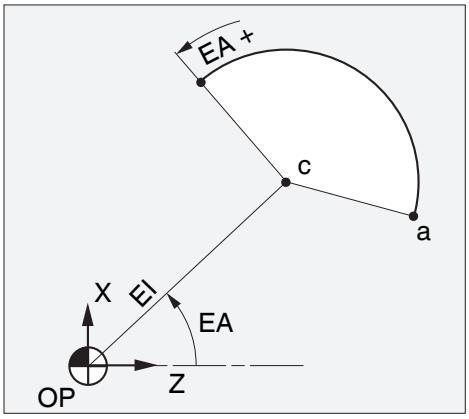
Notes

The notes below concern only circles defined by the arc angle (with the centre defined in polar coordinates). When a value of zero is assigned to EA, the system describes a complete circle. When the circle is defined from Z to X, EA is positive; in the opposite direction, EA is negative.

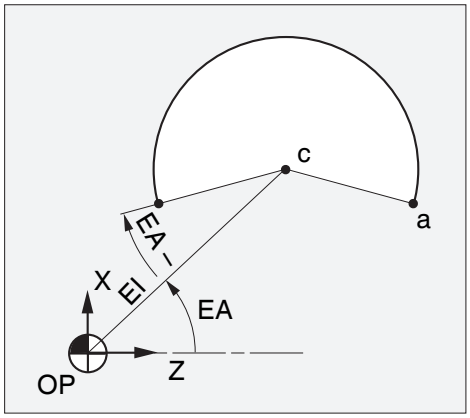
Four types of circles are possible in absolute programming, depending on the direction of the programmed circular interpolation (G02 or G03) and the sign (positive or negative) of arc angle EA.

Example

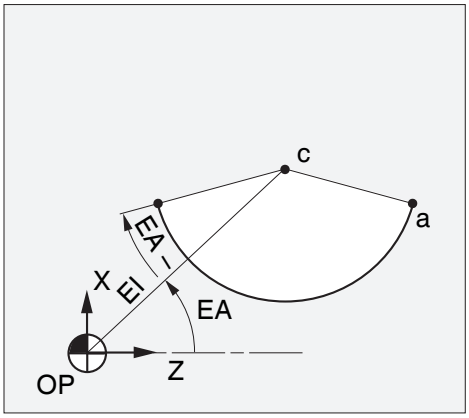
```
N.. G90 G20 Xa Za
N.. G03 EA.. EI.. EA+160
```



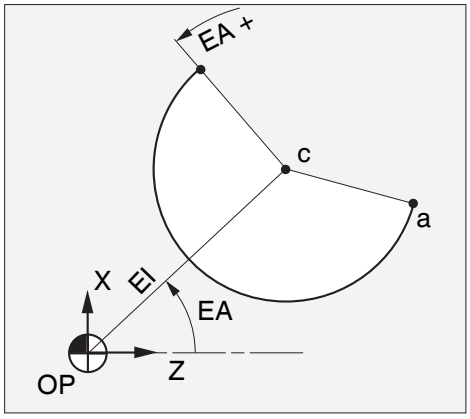

```
N.. G90 G20 Xa Za
N.. G03 EA.. EI.. EA-160 (or EA+200)
```




```
N.. G90 G20 Xa Za
N.. G02 EA.. EI.. EA-160
```




```
N.. G90 G20 Xa Za
N.. G02 EA.. EI.. EA+160 (or EA-200)
```

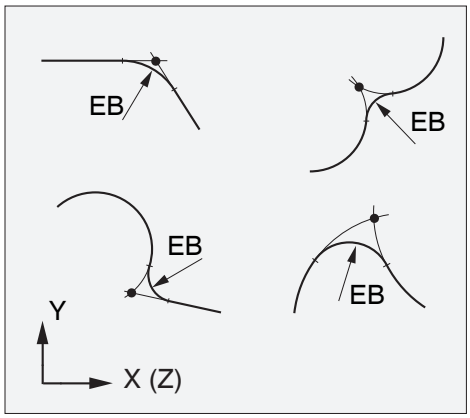


5.7.8 Programming Fillets and Chamfers

5.7.8.1 Fillet between two interpolations ► EB+

EB+	Fillet between two interpolations.

This functions is used to make a fillet between two linear and/or circular interpolations.



Syntax (XY plane)

N.. G01/G02/G03 X.. Y.. I.. J.. / R.. [F..] **EB+..** [EF..]

G01 / G02 / G03	Linear or circular interpolations.
X.. Y..	Programmed intersection point.
I.. J.. / R..	Circle centre or radius in G02 or G03.
F..	Feed rate.
EB+..	Fillet dimension.
EF..	Feed rate specific to the fillet.

Property of the Function

Function EB+.. is nonmodal.

Cancellation

Function EB+ is cancelled at the end of the block.

Syntax (XZ plane)

N.. G01/G02/G03 X.. Z.. I.. K.. / R.. [F..] **EB+..** [EF..]

G01 / G02 / G03	Linear or circular interpolations.
X.. Z..	Programmed intersection point.
I.. K.. / R..	Circle centre or radius in G02 or G03.
F..	Feed rate.
EB+..	Fillet dimension.
EF..	Feed rate specific to the fillet.

Property of the Function

Function EB+.. is nonmodal.

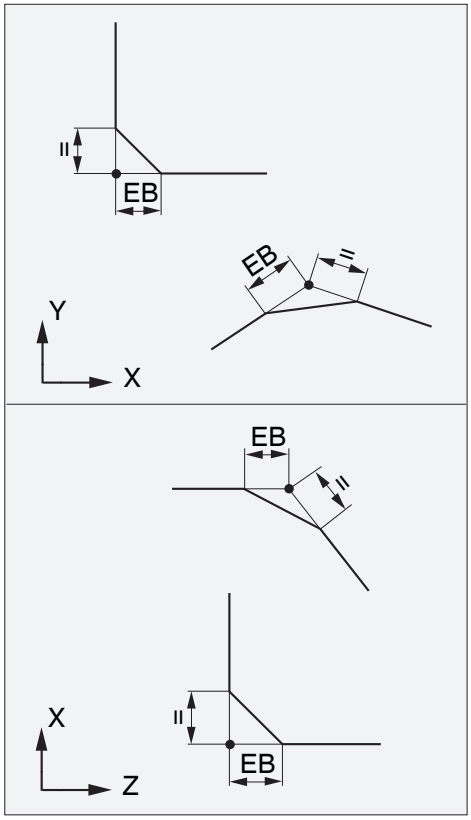
Cancellation

Function EB+ is cancelled at the end of the block.

5.7.8.2 Chamfer Between Two Linear Interpolations ► EB-

EB-	Chamfer between two linear interpolations.

This functions is used to make a chamfer between two linear interpolations.



Syntax (XY plane)

N.. G01 X.. Y.. [F..] **EB**-.. [EF..]

G01	Linear interpolation.
X.. Y..	Programmed intersection point.
F..	Feed rate.
EB-..	Chamfer dimension.
EF..	Feed rate specific to the chamfer.

Property of the Function

Function EB-.. is nonmodal.

Cancellation

Function EB- is cancelled at the end of the block.

Syntax (XZ plane)

N.. G01 X.. Z.. [F..] **EB**-.. [EF..]

G01	Linear interpolation.
X.. Z..	Programmed intersection point.
F..	Feed rate.
EB-..	Chamfer dimension.
EF..	Feed rate specific to the chamfer.

Property of the Function

Function EB-.. is nonmodal.

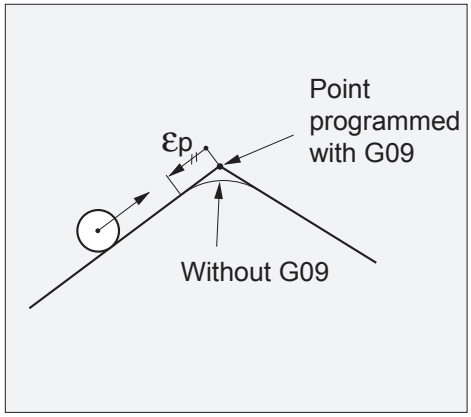
Cancellation

Function EB- is cancelled at the end of the block.

5.7.9 Accurate End of Block Stop ► G09

G09	Accurate stop at end of block before continuation to next block.

When the function is programmed, the CNC system will wait until the axes are precisely at the programmed position before resuming execution.



Syntax

N.. **G09** [G00/G01/G02/G03] X.. Y.. Z.. [F..]

G09	Accurate stop at the end of a block before continuation to the next block.
G00/G01/G02/G03	Linear or circular interpolation.
X.. Y.. Z..	Coordinates of the end point.
F..	Feed rate.

Property of the Function

Function G09 is nonmodal.

Cancellation

Function G09 is cancelled at the end of the block.

Notes

The lag error εp is directly proportional to the feed rate.

The more acute the angle between two paths, the greater the smoothing effect at a given feed rate and therefore a given εp.

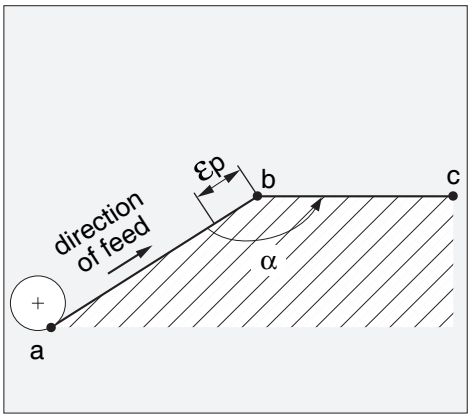
When G09 is programmed:
- the lag εp is decreased up to Zero at the end of the block,
- the feed rate is also zero at the end of the block.

Examples

G09 activated

The axes are decelerated along path ab at a distance εp from point b and go through point b.

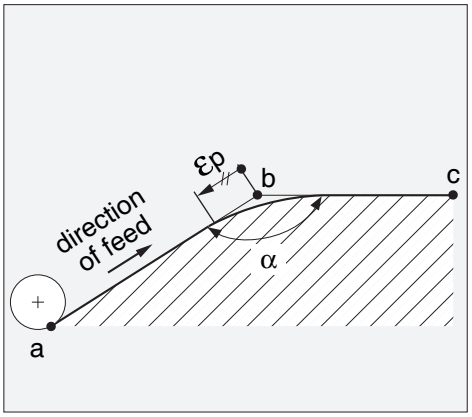
```
N.. ...
N100 G01 Xa Ya F500
N110 G09 Xb Yb
N120 G09 Xc Yc
N..
```



G09 inactive

The axes are not decelerated and do not pass exactly through point b.

The resulting curve between the paths result from the feed rates along ab and bc and the angular value of the vectors.



5.8 «Inclined Axis» or «Inclined Wheel»

General

On a grinding machine, the «inclined axis» or «inclined wheel» state can be declared by programming the corresponding external parameter E :

- parameter E69001 for the «inclined axis» state,
- parameter E 69002 for the «inclined wheel» state.

In both cases, the angle of inclination declared:

- can be positive or negative,
- must be between -80 and +80 degrees (or error message 94 is returned),
- is expressed in ten thousandths of a degree.

It should be noted that the «inclined axis» and «inclined wheel» states cannot be declared together in a channel (otherwise error message 95 is returned) and that the angle of inclination is saved after a reset.

«Inclined Axis» State

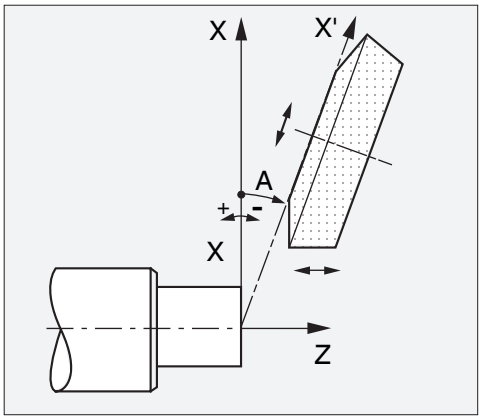
Angle Declaration in the «Inclined Axis» State

The inclination angle can be declared:

- by programming external parameter E69001. Example: E69001 = -200000 (angle = -20 degrees), or,
- Automatically by PLC. (See Machine Operation Manual).

The axis inclination is the value of angle A between the penetration axis along X and the normal to the workpiece-holder spindle.

In the «inclined axis» state, movements programmed on the axis are parallel to the axis inclination angle.



Check of Travels

The travels declared in the machine parameters are relative to the physical axes. With linear or circular interpolation (system in state G20) it is checked that the path does not go beyond the soft limits on the X and Z axes.

Axis Movement

When an axis inclination is entered, the following elements can be defined in cartesian coordinates:

- linear and circular movements (the system performs the conversion of these movements for the servo-system)
- machining parameters such as origin offset, machine travels and tool corrections.

The pivot point of the inclined axis must be referenced to the measurement origin.

Angle Declaration in the «Inclined Wheel» State

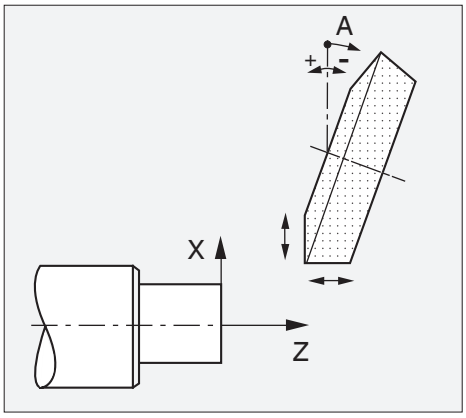
The angle is declared by programming external parameter E69002.

Example:

E69002 = -150000 (angle = -15 degrees).

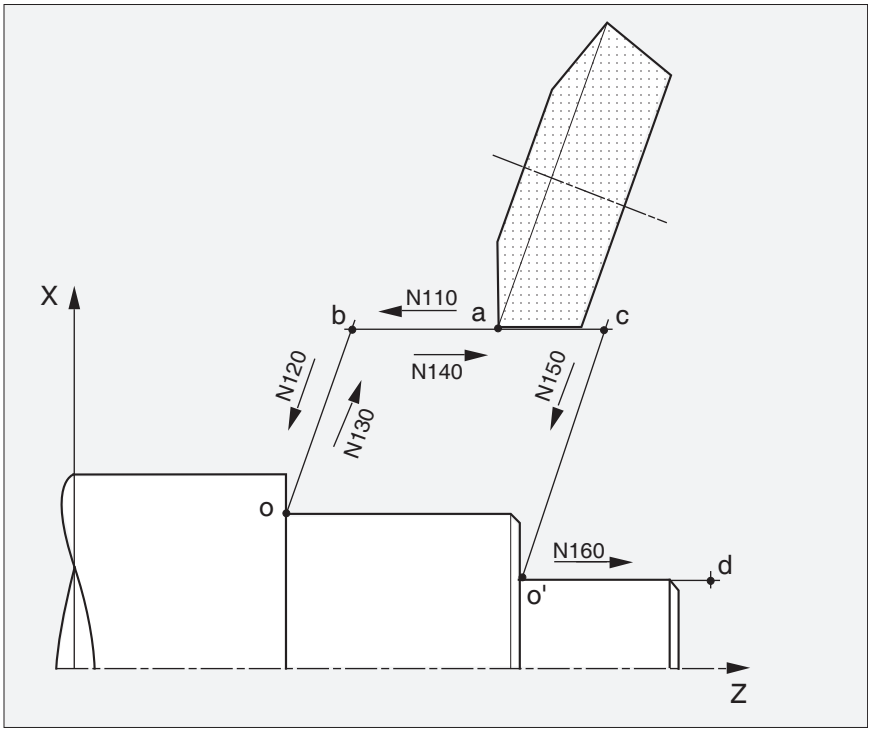
In the «inclined wheel» state, wheel movements programmed on X and Z are orthogonal.

Functions G05 and G07 are used in the same way as in the «inclined axis» state (see example below).



Example

Positioning in Z and X in absolute dimensions (G90) with wheel inclined by -15 degrees.



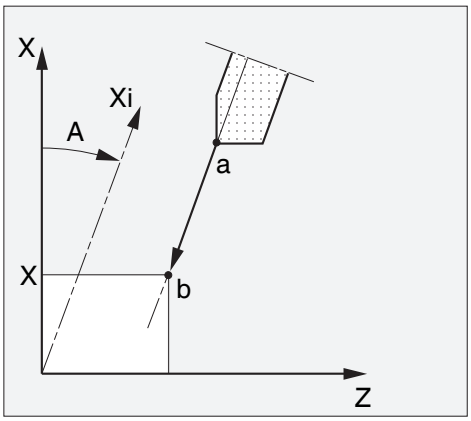

```
N..
N.. E69002=-150000                Wheel inclination by parameter E
N.. G97 S300 M03 M40
N.. G95 F0.1
N..
N100 G00 X50 Z50                  Point a, position before switching to G07
N110 G07 X20 Z25                  Definition of the reference point (o) and positioning in point b
                                  on Z
N120 G01 G05 X20F0.2              Point o, approach along X
N130 G05 X50                      Point b, return along X
N140 G07 X5 Z30                   Definition of the reference point (o') and positioning in point c
                                  on Z
N150 G05 X10                      Point o', approach along X
N160 Z50                          Point d, movement along Z
N..
```

5.9 Initial Tool Positioning Before Machining on an Inclined Axis ► G07

G07	Initial tool positioning before machining on an inclined axis.

This functions defines the location on the Z axis where it intersects with the line parallel to the inclined axis going through the reference point whose position is programmed with the function.

In the figure:
- point a: path start,
- point b: intersection with the line parallel to the inclined axis.



Syntax

N.. [G90] [G00/G01] **G07** X.. Z..

G90	Programming in absolute dimensions (always).
G00/G01	Linear interpolation.
G07	Initial tool positioning before machining on the inclined axis.
X.. Z..	Reference point.

Property of the Function

Function G07 is nonmodal.

Cancellation

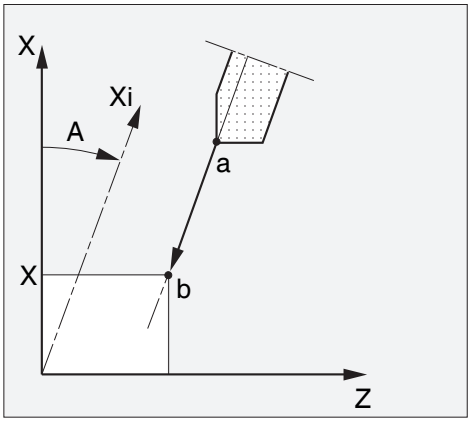
Function G07 is cancelled at the end of the block.

5.10 Movement Along the Inclined Axis ► G05

G05	Movement along the inclined axis.

This function defines movement on the X axis parallel to the inclined axis up to the cartesian end point programmed.

In the figure:
- point a: path start point,
- point b: end point.



Syntax

N.. [G90/G91] [G00/G01] **G05** X..	
G90/G91	Programming in absolute or incremental dimensions.
G00/G01	Linear interpolation.
G05	Programming of a movement along the inclined axis.
X..	End point.

Property of the Function

Function G05 is nonmodal.

Cancellation

Function G05 is cancelled at the end of the block.

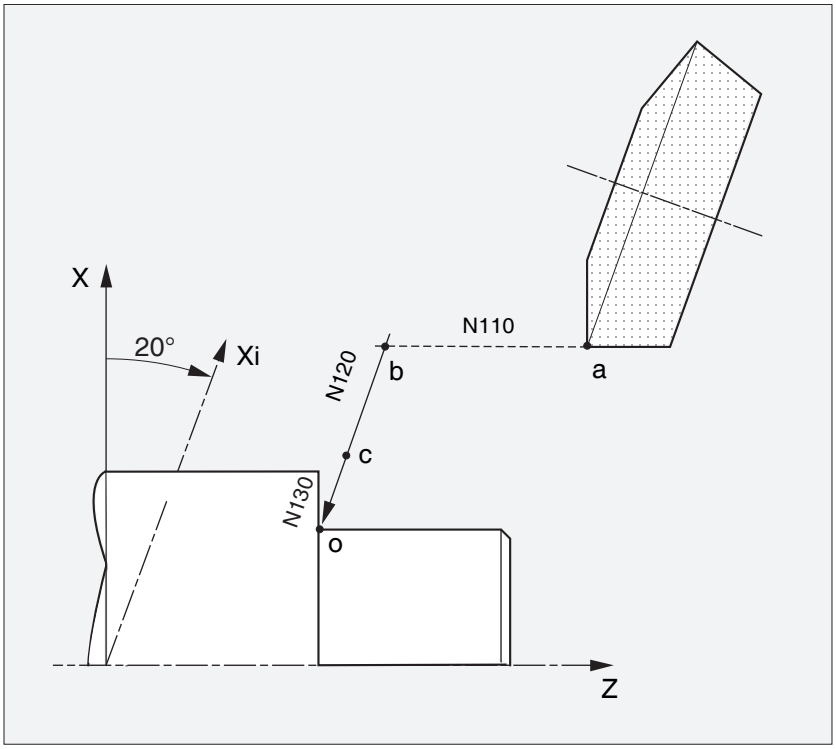
Notes

Function G07 must always be programmed before function G05.

For functions G05 and G07, failure to comply with the following programming rules cause generation of error message 7:
- the functions must be programmed in plane G20,
- the interpolation must be programmed by G00 or G01,
- function G07 must be followed by X and Z,
- function G05 must be followed by X.

Example

Positioning in Z and X in absolute dimensions (G90) with axis inclined by -20 degrees.




```
N..
N.. E69001=-200000
N.. G97 S300 M03 M40
N.. G95 F0.1
N..
N.. G00 X60 Z70
N110 G07 X10 Z25

N120 G05 X30
N130 G01 G05 X10 F0.2
N..
```

Axis inclination by parameter E

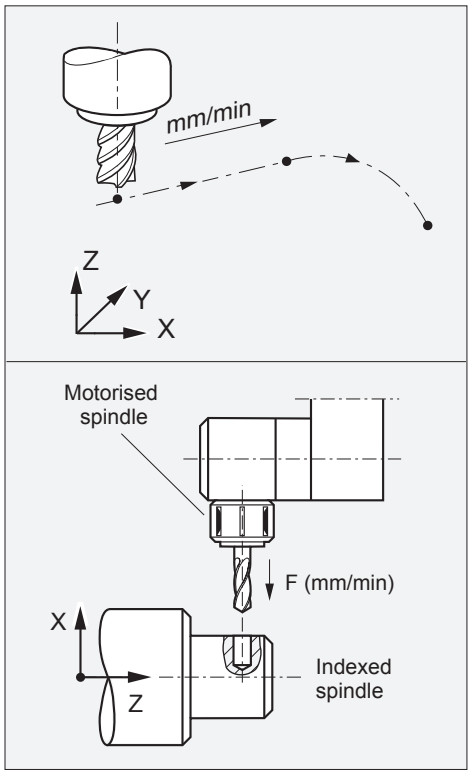
Point a, position before switching to G07
Definition of the reference point (o) and positioning in point b on Z
Point c, positioning on X
Point o, positioning on X

5.10.1 Feed Rate

5.10.1.1 Feed Rate Expressed in Millimetres, Inches or Degrees per Minute ► G94

G94	Feed rate expressed in millimetres, inches or degrees per minute.

The feed rate is expressed in millimetres or inches per minute on linear axes and in degrees per minute on programmed rotary axes alone.



Syntax

N.. **G94** F.. G01/G02/G03 X.. Y.. Z.. A.. B.. C..

G94	Function setting the feed rate: - in millimetres/min, - in inches/min, - in degrees/min.
F..	Mandatory argument associated with the function and defining the feed rate.
G01/G02/G03	Linear or circular interpolation.
X.. Y.. Z..	End point on linear axes.
A.. B.. C..	Angular end point on rotary axes.

Properties of the Functions

Function G94 is modal. It is the default function.

Reminder

A default value of 1000 mm/min or 100 Inches/min is assigned to address F at power on.

Cancellation

Function G94 is cancelled by one of functions G93 and G95.

Notes

The feed rate limits are defined by the machine manufacturer (see manufacturer's technical data). When the programmed feed rate exceeds the permissible minimum or maximum rate, the system automatically limits it to the permissible feed rate.

To enter a feed rate in inches/min, the function G70 must be active.

When switching from G94 (unit/min.) to G95 (units /rev.) or back, new G code and F value must be programmed together in the block.

Example

```
N..
N140 G00 X.. Y..
N150 G95 F0.3 G01 Z..
N160 X.. Z.. F0.2
N..
N240 G00 X.. Y.. Z..
N250 G94 F200 G01 Y.. W..
N260 W.. F100
N..
```

Feed rate in mm per revolution

Feed rate in mm/min on a primary axis and a secondary axis

5.10.1.2 Programming additional axes

When primary axes (X, Y, Z) are programmed, the feed rate is applied on the trajectory they define. Additional axes will follow so that they execute the motion in the same time as the primary axes.

When secondary (U,V,W) and rotary (A, B, C) axes are programmed together the feed rate is applied on the pseudo trajectory defined by U,V, W and the rotary axes will follow as stated above.

5.10.1.3 Programming of a modulo rotary axis programmed alone

Example:

```
N.. G91 G94 F40 G01 B30
```

Feed rate F40 is expressed in degrees per minute (execution time = 45 seconds).

5.10.1.4 Programming of modulo rotary axis and a linear axis

Example:

```
N.. G91 G94 F100 G01 X10 B30
```

The feed rate on the X axis is expressed in millimetres per minute. The feed rate on the B axis depends on the time required to execute the linear path on the X axis.

$t = \Delta X / F = 10 / 100 = 0.1$ minute, i.e. 6 seconds.

The feed rate on the B axis is equal to 30 deg/6 seconds, i.e. 5 deg/s.

5.10.1.5 Programming of two modulo rotary axes

Example:

```
N.. G91 G94 F100 G01 A30 B40
```

The feed rates on A and B axes are expressed in degrees per minute.

Execution time : $t = \sqrt{\Delta A^2 + \Delta B^2} / F$ i.e. t = 30 seconds.

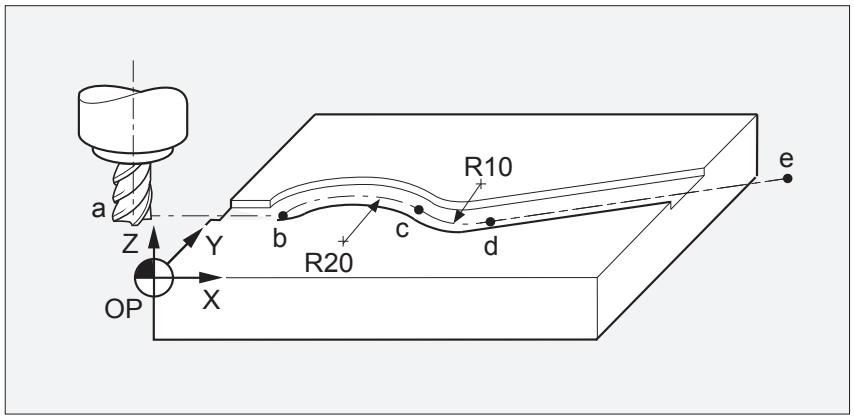
The feed rate on each axis is equal to:

- Feed rate on A = $\Delta A/t$, i.e. 60 deg/min
- Feed rate on B = $\Delta B/t$, i.e. 80 deg/min

5.10.1.6 C axis feed rate for Turning Machine

When the C axis is programmed the feed rate represents the relative speed at the tool tip position. Therefore the C rotation speed depends of the position of the X axis.

Example



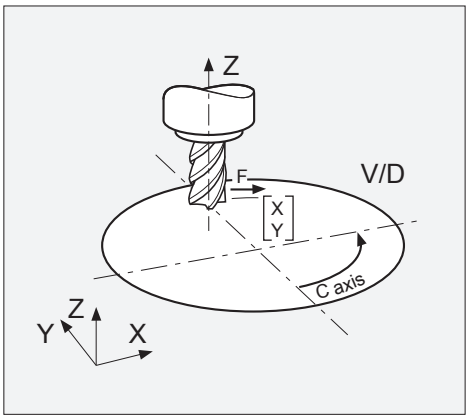

```
%40
N10 G00 G52 Z..                    Rapid to tool change
N20 ...                            Tool call
N30 S800 M40 M03
N40 G94 F100                       Feed rate in mm/min
N50 G00 Xa Ya Z-5
N60 G01 Xb
N70 G02 Xc Yc R20 F160             Change in feed rate
N80 G03 Xd Yd R10 F70              Change in feed rate
N90 G01 Xe Ye F100                 Change in feed rate
N..
```

5.10.2 Inverse Time Feed Rate Coding (V/D) ► G93

G93	Feed rate in invert of time (1/min.)

The feed rate is programmed in inverse time when the numerical control cannot calculate the length of a path and therefore a feed in inc/min. makes no sense.

Example: A rotary axis programmed alone or with linear axes.



Syntax

N.. **G93** F.. G01 X.. Y.. Z.. A.. B.. C..

G93	Function setting inverse time feed rate coding: min-1.
F..	Mandatory argument associated with the function defining the feed rate.
G01	Linear interpolation at the programmed feed rate.
X.. Y.. Z..	End point on the linear axes.
A.. B.. C..	Angular end point on the rotary axes.

Property of the Function

Function G93 is modal.

Cancellation

Function G93 is cancelled by one of functions G94 and G95.

Notes

The feed rate limits are defined by the machine manufacturer (see manufacturer's technical data). When the programmed feed rate exceeds the permissible minimum or maximum rate, the system automatically limits it to the permissible feed rate.

Reminder

$$\text{Feed rate } F = \frac{V \leftarrow \text{feed rate in mm/min}}{D \leftarrow \text{Path length in mm}}$$

Inverse time feed rate should not be used for circular and helical interpolation (no error message).

When the feed rate unit is changed, the G.. function defining the new feed rate unit must be followed by argument F (if the system is already in G93 mode, the F.. address can be programmed alone in a block).

Example

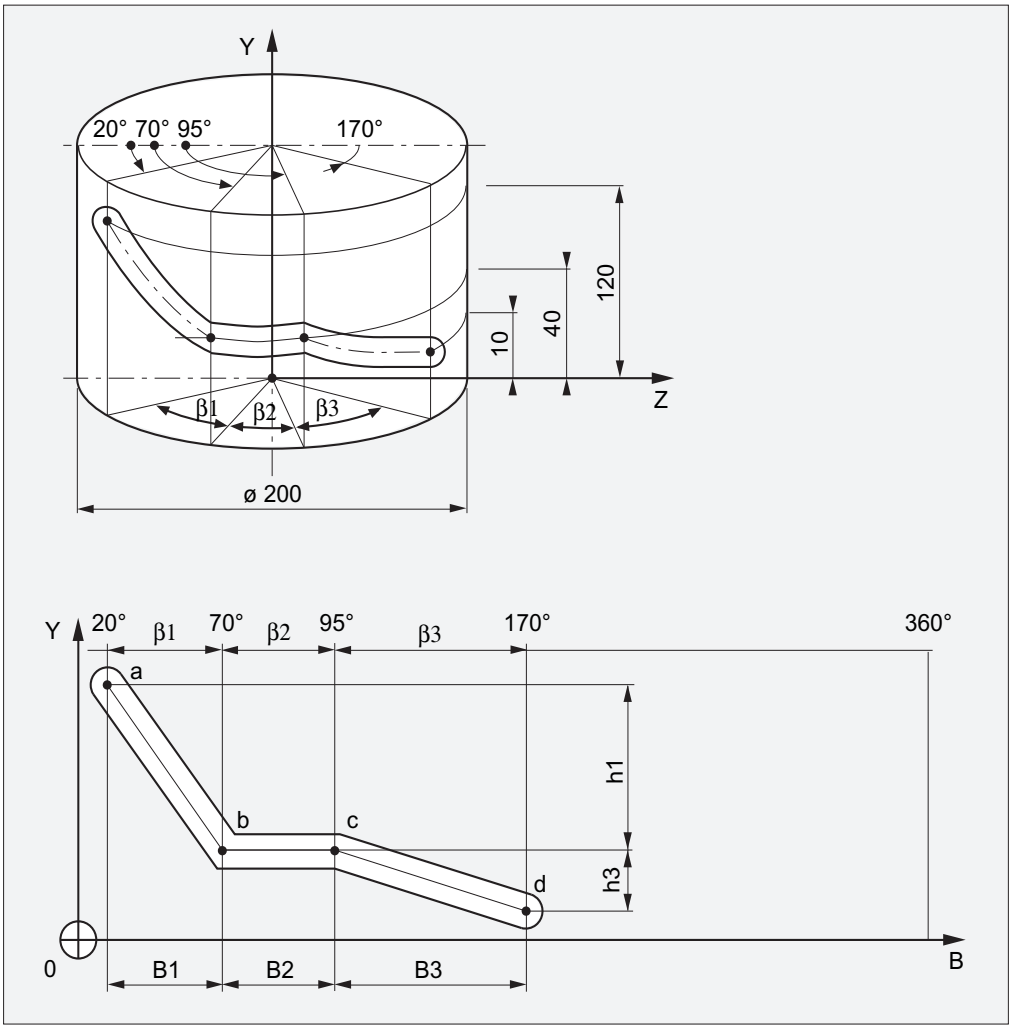
```
N..
N140 G00 X.. Y.. Z..
N150 G94 F200 G01 X.. Y..          Feed rate in mm/min
N160 Y.. F100
N..
N210 G00 Y.. Z..
N220 G93 F50 G01 U.. C..           Feed rate in V/D on a secondary axis and a rotary axis
N.. C.. F30
N..
```

Example

F determination in G93

Groove depth = 5 mm. Part diameter = 200 mm

Required cutting speed (V) = 150 mm/min



	Length in mm of the developed arc	Real arc length L in mm	Feed rate in V/D
	$B = \frac{\pi \times D \times \beta}{360}$	$L = \sqrt{B^2 + h^2}$	F = V/D
a b	B1 = 87.266	L1 = 118.386	F1 = 1.26
b c	B2 = 43.633	L2 = 43.633	F2 = 3.43
c d	B3 = 130.899	L3 = 134.293	F3 = 1.12

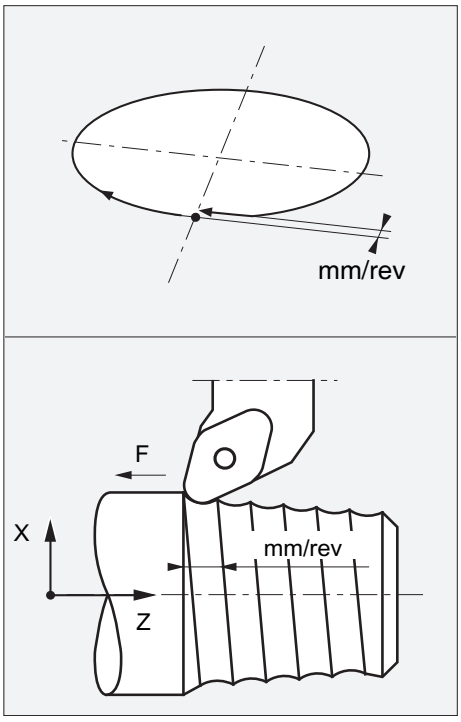
```
%50
N10 (G94)                    Feed rate initialised in mm/min
N20 ...                      Tool call
N30 S500 M40 M03
N50 G00 X0 Z150 B0           Approach position
```

```
N60 Y120 Z102 B20          Point a
N70 G01 Z95 F150           Penetration in Z
N80 G93 F1.26 Y40 B70      Point b, feed rate in V/D
N90 B95 F3.43              Point c, feed rate in V/D
N100 Y10 B170 F1.12        Point d, feed rate in V/D
N110 G00 Z300              Retraction on Z
N120 Y300 B0 M05
N130 M02
```

5.10.3 Feed Rate Expressed in Millimetres or Inches per Revolution ► G95

G95	Feed rate expressed in millimetres or inches per revolution.

The feed rate is expressed in millimetres or inches per spindle revolution.



Syntax

N.. **G95** F.. G01 / G02 / G03 X.. Y.. Z..

G95	Function setting the feed rate: - in mm/rev, - in in./rev.
F..	Mandatory argument associated with the function defining the feed rate.
G01 / G02 / G03	Linear or circular interpolation at the programmed feed rate.
X.. Y.. Z..	End point on the linear axes.

Property of the Function

Function G95 is modal.

Reminders

- G94 (mm/min) is the default function.
- A default value of 1000 mm/min is assigned to address F (F1000).

Cancellation

Function G95 is cancelled by one of functions G93 and G94.

Notes

The feed rate limits are defined by the machine manufacturer (see manufacturer's technical data). When the programmed feed rate exceeds the permissible values, the system automatically limits it. In addition, the system limits the feed rate to 30 mm/rev anyway.

If a higher value is programmed, the system does not return an error message but simply limits the feed rate to 30 mm/revolution).

When the feed rate unit is changed, the G function defining the new feed rate unit and its programming format must be followed by the argument F.. (if the system is already in G95 mode, address F.. can be programmed alone in a block).

Example

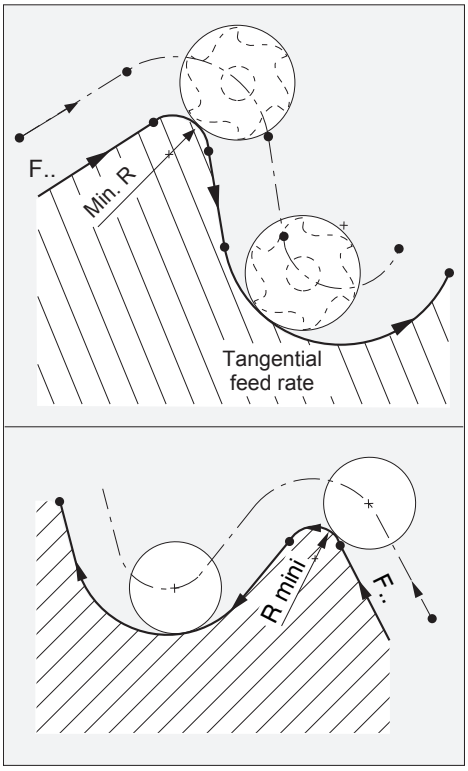
```
N..
N.. G00 X.. Y..
N140 G94 F200 G01 Z..          Feed rate in mm/min
N150 X.. Y.. F100
N..
N240 G00 X..  Y..
N250 G95 F0.3 G01 X.. W..      Feed rate in mm/rev on a primary axis
                               and a secondary axis
N260 W.. F0.2
N..
```

5.10.4 Tangential Feed Rate ► G92 R..

G92 R..	Programming the tangential feed rate

The feed rate is normally applied at the tool center.

With this function it is possible to choose to apply the feed rate along the part when using tool radius correction.



Syntax

N.. **G92** R..

G92	Tangential feed rate applied to tool radius correction.
R..	Mandatory argument defining the minimum value of curve radius below which the tangential feed rate calculation (which could otherwise reach a too high value) is to be ignored.

Properties of the Function

Function G92 followed by argument R is modal.

Cancellation

Tangential feed rate G92 R.. is cancelled by:
- cancellation function G92 R0,
- function G92 R.. with a different radius,
- the end of program function (M02),
- a reset.

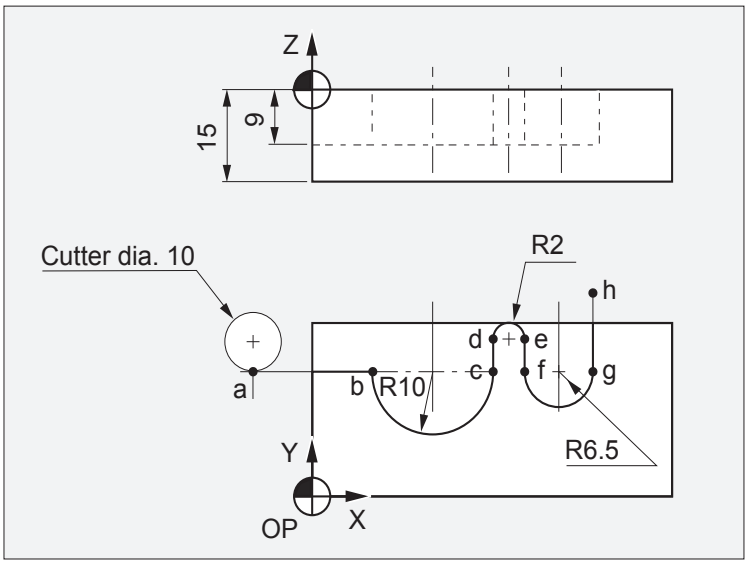
Notes

Function G92 is ignored during automatic creation of a connecting circle between two secant elements (lines or circles) with radius correction. The feed rate will then be the same as the feed rate programmed in the previous block.

Function G92 programmed in a block must not be accompanied by axis commands.

Example

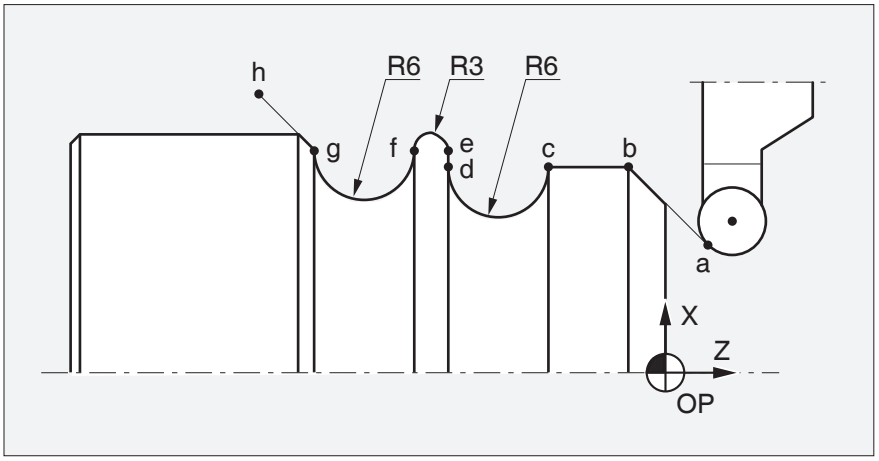
In this example, the tangential feed rate is applied to curves whose radius is greater than 3 mm.




```
%22
N10 G00 G52 Z..                          Retraction for tool change
N20...                                   Tool call
N20 T12 D12 M06 (CUTTER DIA=10)
N30 S600 M40 M03
N40 G00 G41 Xa Ya                        Point d, left radius offset
N50 Z-9
N60 G92 R3                               Tangential feed rate limit
N70 G01 Xb F200
N80 G03 Xc Yc R10                        Feed rate applied to point of tangency
N90 G01 Yd
N100 G02 Xe Ye R2                        Feed rate applied to tool centre
N110 G01 Yf
N120 G03 Xg Yg R6.5
N130 G01 Yh
N140 G92 R0                              Tangential feed rate cancelled
N150 G00 G40 G52 Z..
N160 M02
```

Example

In this example, the tangential feed rate is applied to curves whose radius is greater than 4 mm.



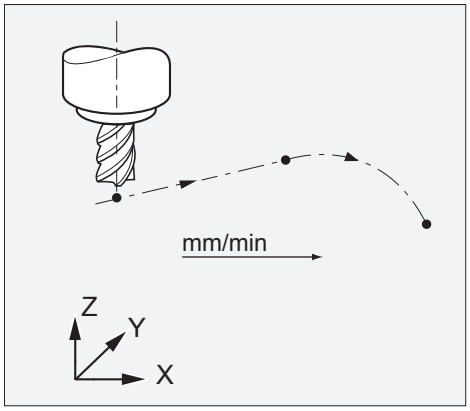

```
%23
N10 G00 G52 X.. Z..                          Tool change position
N20 _                                        Tool call
N30 S900 M40 M04
N40 G95 F0.2
N50 G00 G42 Xa Za                            Right radius offset
N60 G92 S3000
N70 G96 S100
N80 G92 R4                                   Tangential feed rate limit
N90 G01 Xb Zb F0.2
N100 Xc Zc
N110 G02 Xd Zd R6                            Feed rate applied to the point of tangency
N120 G01 Xe, Ze
N130 G03 Xf Zf R3                            Feed rate applied to the tool centre
N140 G02 Xg Zg R6
N150 G01 Xh Zh
N160 G92 R0                                  Tangential feed rate cancelled
N170 G00 G40 G52 X.. Z.. G97 S900 M05
N180 M02
```

5.10.5 Setting feed rate per axis set ► G92 F

G92 F Setting feed rate per axis

For a sequence of blocks containing movements on linear and rotary axis G92 F... allows to set the feed rate on one or more axes but not necessarily all axes involved in the trajectory.

This is generally used to maintain a constant speed on rotary axes.



Syntax

G92 F... <list of axes>

G92	Introduces the function feed rate per axes set.
F..	Feed rate to use.
<[X][Y][Z][...][...]>	Set of axes on which combined movement the feed rate <will be applied

Properties of the Function

Function G92 F is modal
Default is Feed rate applied on the primary axes.

Cancellation

- G92 F... <other set of axes>
- G92 F no axes
- Reset

When the argument F is programmed after function G92, the block includes a list of axes on which the feed rate F is to be applied. If no axes are listed, the feed rate is processed as usual i.e. to the primary axes.

Errors detected

The system generates error message 2 in the following cases:

- The axes list includes a value,
- Some axes from the list do not belong to the group,
- The list of axes includes both linear and rotary axes.

Example

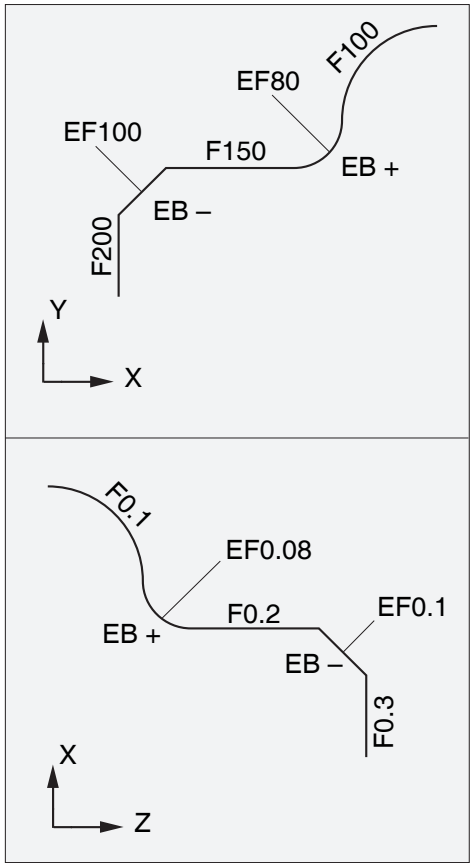
G92 F50 A

A+50 X1	Axis A will move at 50 deg/min the speed of X axis will be adapted to follow
A+75 X10	

5.10.6 Feed Rate Specific to Fillets EB+ and Chamfers EB- ► EF

EF	Feed rate specific to fillets EB+ and chamfers EB-.

A feed rate different from the modal machining feed rate F can be programmed for fillets and chamfers programmed by EB+ and EB-.



Syntax

N.. Interpolation EB+.. / EB-.. **EF..**

Interpolation	Linear interpolation (G01) or circular interpolation (G02 or G03).
EB+	Fillet dimension.
EB-	Chamfer dimension.
EF..	Feed rate.

Property of the Function

Function EF.. is modal.

Cancellation

Function EF followed by a value is cancelled by programming:

- function EF followed by a new value, or
- the end of the program (M02).

Notes

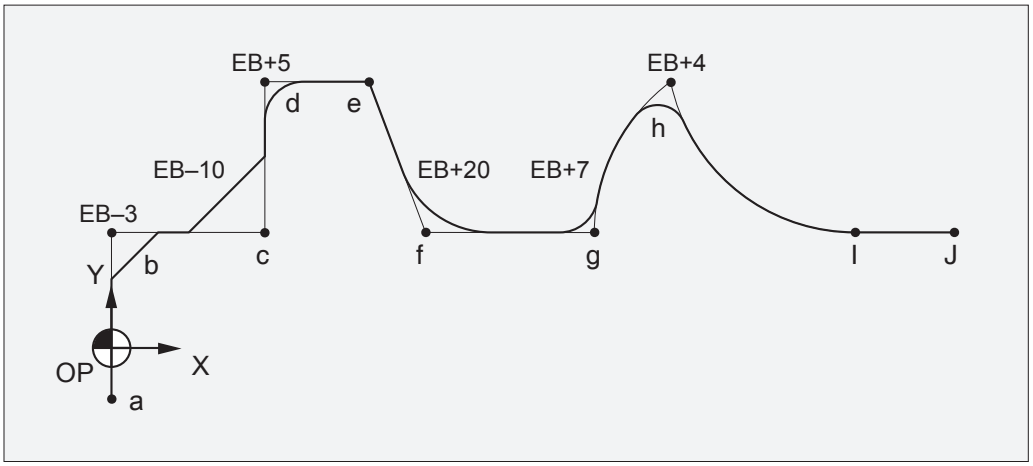
Feed rate EF is substituted for the modal feed rate F if its value is nonzero and is less than feed rate F.

Feed rate EF is in the unit specified by G94 (mm/min) or G95 (mm/rev).

Feed rate F.. in mm/min (G94) or mm/rev (G95) remains modal when executing fillets and/or chamfers.

Example

Undimensioned radii = cutter radius

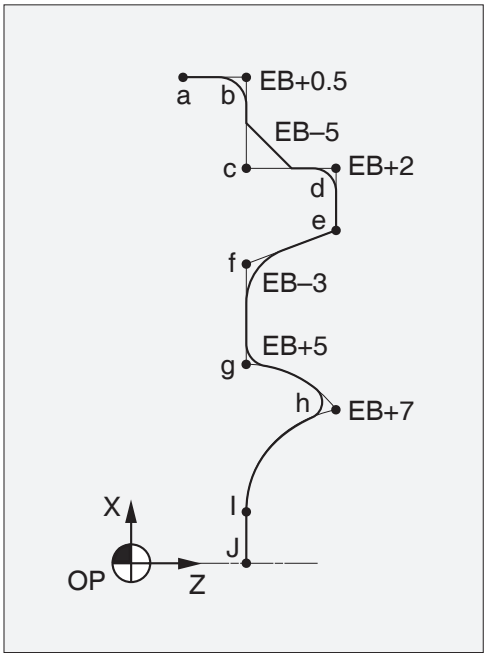


When executing the contour, the linear and circular interpolations are carried out at modal feed rate G94 F120.

```
%37
N10 G00 G52 Z..                    Tool spindle positioning
N20 ...                            Tool call
N30 S800 M40 M03
N40 G92 R1
N50 X0 Y-10                        Point a
N60 Z-5
N70 G94 F120                       Feed rate for G01, G02 or G03 interpolations
N80 G1 Y15 EB-3 EF90               Point b (feed rate  90)
N90 X20 EB-10 EF70                 Point c (feed rate  70)
N100 Y35 EB+5                      Point d (feed rate  70)
N110 X40                           Point e (feed rate  120)
N120 X50 Y15 EB+20 EF90            Point f (feed rate  90)
N130 X70 EB+7 EF70                 Point g (feed rate  70)
N140 G02 X80 Y35 R40 EB+4          Point h (feed rate  70)
N150 G03 X100 Y15 R30              Point i (feed rate  120)
N160 G1 X120                       Point j (feed rate  120)
N170 ...
```

Example

Finishing a contour with feed rate EF in the chamfers and fillet



When executing the contour, the linear and circular interpolations are carried out at modal feed rate G95 F0.4.

```
%39
N.. ...
N110 G00 G52 X.. Z.. G97 S900 M04      Tool spindle positioning
N120 ...                               Tool call
N130 X160 Z30                          Point a
N140 G96 S400
N150 G92 S4000
N160 G95 F0.4                          Feed rate for G1, G2 or  G3
N170 G01 Z50 EB+0.5 EF0.3              Point b (feed rate  0.3)
N180 X140 EB-5 EF0.2                   Point c (feed rate  0.2)
N190 Z60 EB+2                          Point d (feed rate  0.2)
N200 X110                              Point e (feed rate  0.4)
N210 Z60 EB-4 EF0.15                   Point f (feed rate  0.15)
N220 X90 Z50 EB+5                      Point g (feed rate  0.15)
N230 X70 EB+5 EF0.2                    Point h (feed rate  0.2)
N240 G02 X40 Z70 R30 EB+7              Point i (feed rate  0.2)
N250 G03 X10 Z60 R20                   Point j (feed rate  0.4)
N260 G1 X0                             Point k (feed rate  0.4)
N270 ...
```

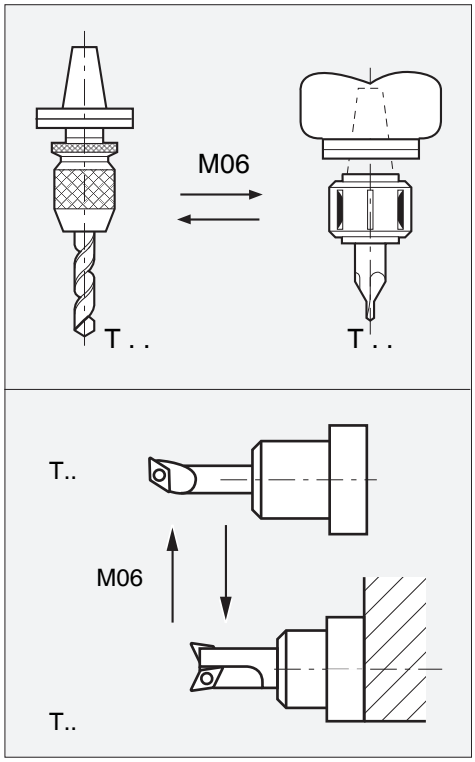
5.11 Tool Programming

5.11.1 Tool Change ► M06

M06	Tool change

This function calls for a tool to be placed in the spindle.

The tool is placed in the spindle either automatically or manually.



Syntax

N.. T.. **M06** [$0 / (...)]

T..	The tool is selected by function T followed by a number. The number corresponds to the location of the tool in the machine magazine.
M06	Tool change.
$0 or (...)	Possible message or comment concerning the tool characteristics.

Properties of the Function

Function M06 is a nonmodal «post» function.

Cancellation

Function M06 is reset at the end of the block.

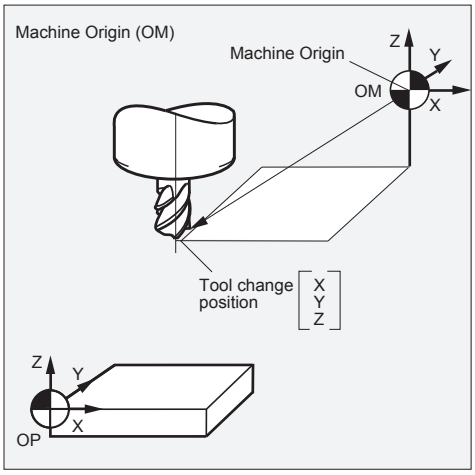
Notes

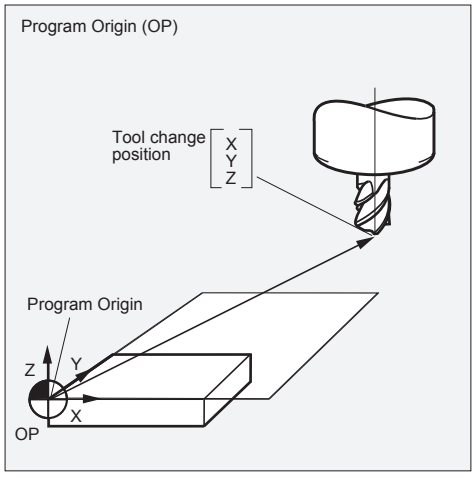
Function T defining the tool number must not be assigned a value greater than 99999999. Above, the system returns error message 1.

Before a tool change, it is recommended to program a safe position for placing the tool in the spindle:

- with reference to the program origin (OP), or
- with reference to the measurement origin programmed with function G52.

Examples






```
N..
N120 G00 G52 Z.. or G00 Z..
N130 T09 M06 (CUTTER DIAMETER=25)
N..
```

Automatic tool change

```
N..
N100 T05 M06 (CUTTER DIAMETER=20)
N..
```

Tool change by subroutine

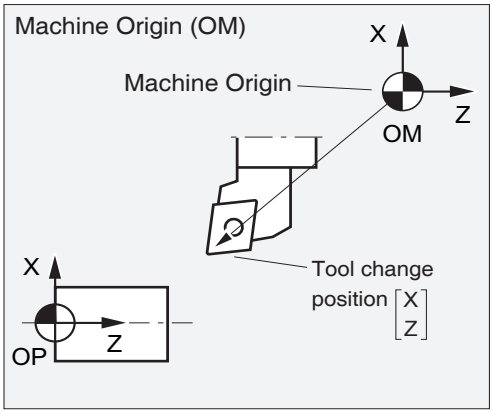
```
N..
N80 T06 (CUTTER DIAMETER=50)
N90 G77 H9000
N..
```

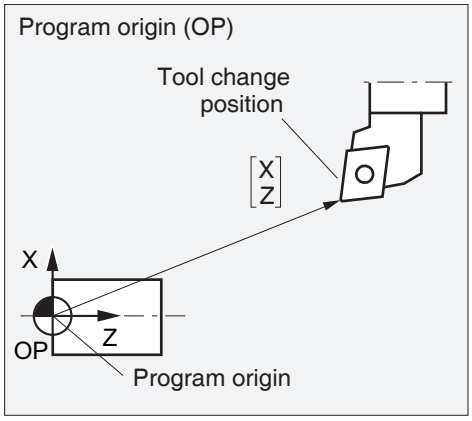
Tool change with spindle preselection and indexing

```
N..
N20 M19
N30 T02 M60 (CUTTER DIAMETER=30)
N40 M06 D02
N..
```

Examples

Automatic tool call





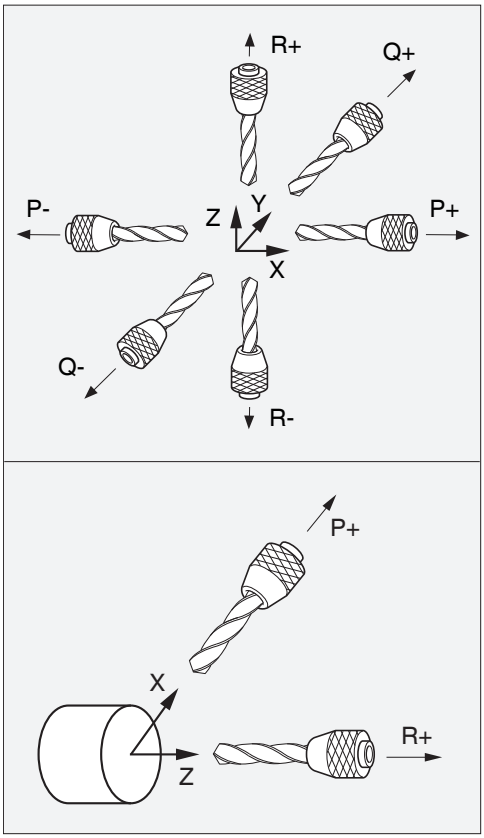

```
N.. ...
N90 G00 X.. Z.. or G00 G52 X.. Z..
N100 T05 M06 (TURN/FACING TOOL R=0.8)
N..
```

5.11.2 Tool Axis Orientation ► G16

G16	Definition of the tool axis orientation with addresses P, Q, R.

The tool axis orientation is defined by this function with one of the compulsory arguments P, Q or R followed by a plus or minus sign.

The tool axis can be oriented in six different positions on machines with an interchangeable head or a right angle attachment.



Syntax

N.. **G16** P±/Q±/R±

G16	Definition of the tool axis orientation.
P+	Points along X+.
P-	Points along X-.
Q+	Points along Y+.
Q-	Points along Y-.
R+	Points along Z+.
R-	Points along Z-.

Properties of the Function

Function G16 followed by one of its arguments P, Q, R is modal.

Function G16 followed by R+ is the default function.

Cancellation

Function G16 followed by one of arguments (P, Q or R) different from the one already programmed cancels the former state G16.

Notes

By convention, the tool vector points from the tool tip (cutting part) to the tool reference (spindle mounting).

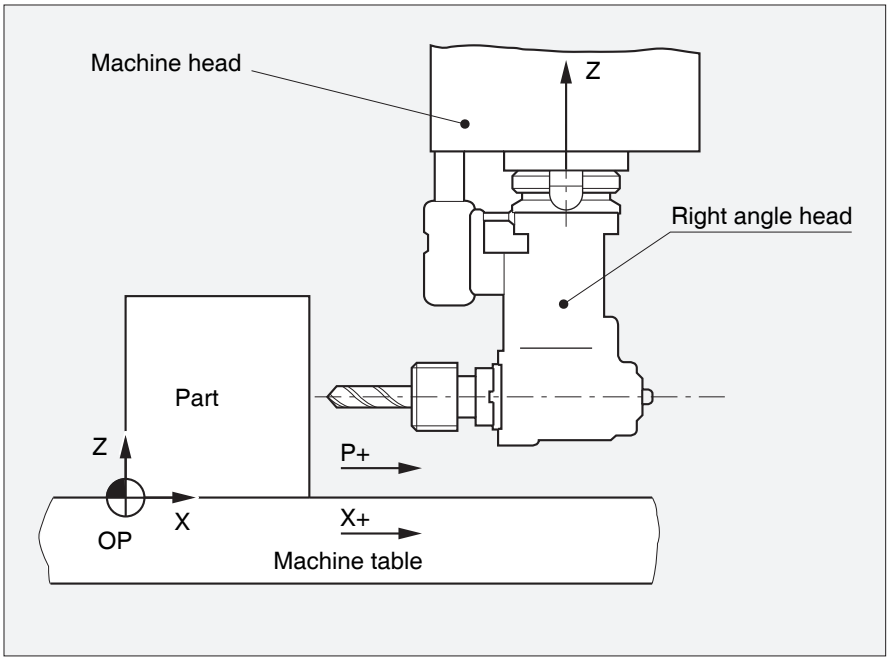
The tool axis cannot be an independent secondary axis.

When defining the tool axis orientation:

- it is recommended to cancel tool radius offset (G40) and canned cycles (G80),
- the block containing G16.. may include movements, miscellaneous M functions and technological S and T functions.

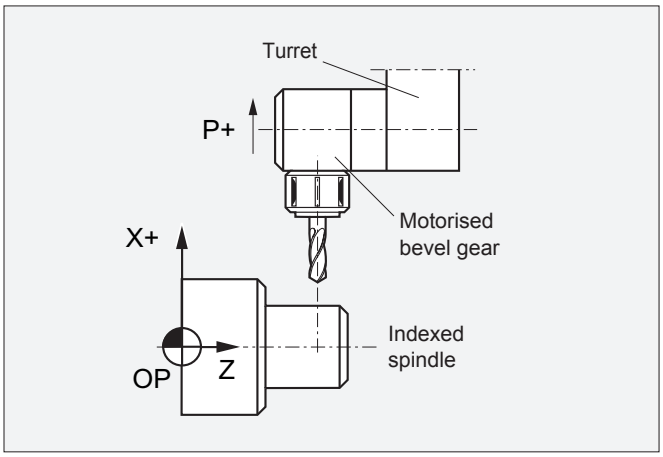
Example

Machine equipped with a right angle head




```
%44
N10 G00 G52 Z.. (G17 G16 R+)        Orientation initialised along Z+
N20 T08 ... M06                     Tool change
N30 S400 M40 M03
N..
N..
N170 G00 G52 X.. Y.. Z..
N180 G16 P+                         Axis orientation along X+
N190 G00 Y.. Z..
N200 G01 X.. F..
N..
```

Example
Machine equipped with a live tool




```
%44
N10 G00 G52 X.. Z.. (G16 R+)          Orientation initialised along Z+
N20 T08 ... M06 (TOOL)
N30 G97 S900 M40 M04
N..
N..
N170 G00 G52 X.. Z..
N180 T11 ... M06 (DRILL)
N190 G97 S200 M03 M40
N200 C0 M19
N210 G16 P+                           Axis orientation along X+
N220 ...
N..
```

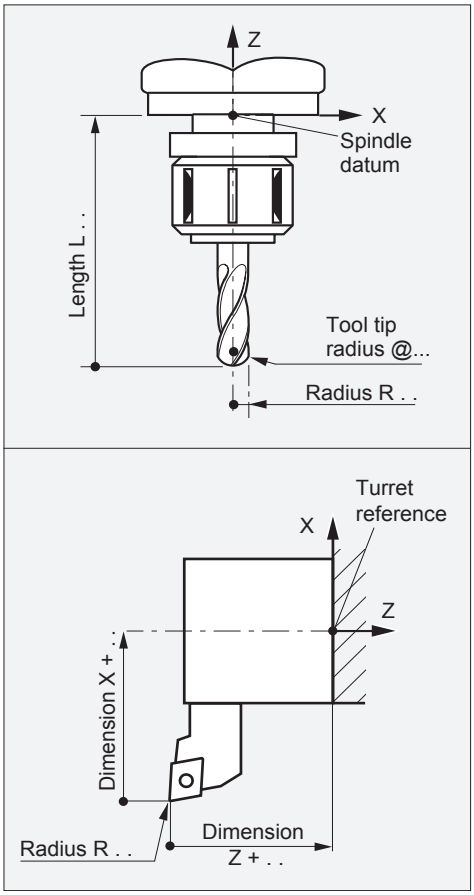
5.11.3 Tool Offset activation ► D

D..	Tool Offset activation.

Address D followed by a number selects the tool correction.

The stored tool dimensions are validated for the programmed axes.

The tool dimensions are displayed as a triplet of corrections on the Context «Tool».



Milling Machine data
- L = Tool length
- R = Tool radius
- @ = Tool tip radius.

The dimensions can be entered:
- manually or via a peripheral,
- by parametric programming.

Turning Machine data
- X = Tool dimension along X axis
- Z = Tool dimension along Z axis
- R = Tool radius
- C = Tool tip orientation

The dimensions can be entered:
- manually or via a peripheral,
- by parametric programming.

Syntax (G17 plane)

N.. [G17] [G16 R+] **D..** [G40/G41/G42] X.. Y.. Z..

G17	XY plane selection.
G16R+	Tool axis orientation along Z+.
D..	Correction number (1 to 255 corrections).
G40	Tool radius offset cancel.
G41/G42	Tool radius offset.
X..Y.. Z..	End point.

Properties of the Function

Function D.. is modal. Correction D0 is the default correction.

Cancellation

Function D.. is cancelled by programming a new correction or D0.

Notes

The correction number may be different from the tool number.

Several correction numbers can be assigned to the same tool.

D0 always contains zero.

The system includes 255 correction correctors. If the number assigned to the correction is higher than 255, the system returns error message 8.

5.11.3.1 Tool length correction (L)

The tool length correction is assigned to the tool axis orientation defined by G16...

The tool length declared is taken into account by programming:

- a correction number D...,
- a movement on the axis parallel to the tool axis orientation.

During machining, tool length settings are re-applied during:

- correction number changes,
- the use of wear corrections,
- a tool axis orientation change.

The length correction is suspended by programming G52.

The maximum dimension of L corrections is 9999.999 mm.

5.11.3.2 Tool dimensions corrections

The X and Z axes (U and W if carried) are affected by the tool dimensions stored in corrections X and Z.

The declared tool dimensions are taken into account when programming:
- a correction number D..,
- a movement of the axis parallel to the tool axis orientation.

During machining, the tool dimensions are re-applied during:
- correction number changes,
- the use of wear corrections,
- a tool axis orientation change.

The corrections are suspended by programming G52.

The maximum dimension of the X and Z corrections is 9999.999 mm.

5.11.3.3 Tool radius corrections

The tool radius correction is assigned to one of the interpolation planes.

The tool radius declared is taken into account when programming:
- the correction number D..,
- one of functions G41 or G42,
- one of the axes of the interpolation plane.

During machining, a change in the tool radius is made by cancelling the radius offset by G40 then reprogramming the radius offset by G41 or G42 after:
- changing the correction number,
- changing the tool wear compensation.

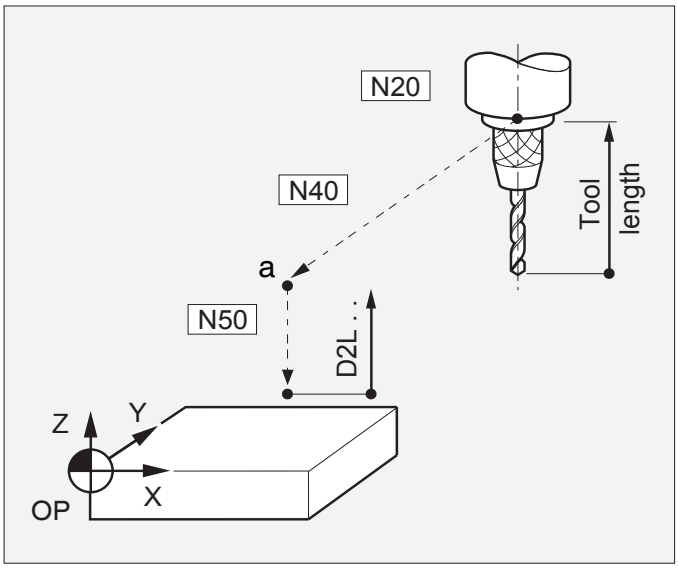
The maximum dimension of R correction is 9999.999 mm.

The two axes of the interpolation plane can be primary axes and carried or independent secondary axes.

Example

Multiple length corrections for a single milling tool

Length corrections L.. of tool T02 are taken into account during the first movement on the Z axis programmed after D02 and D12.

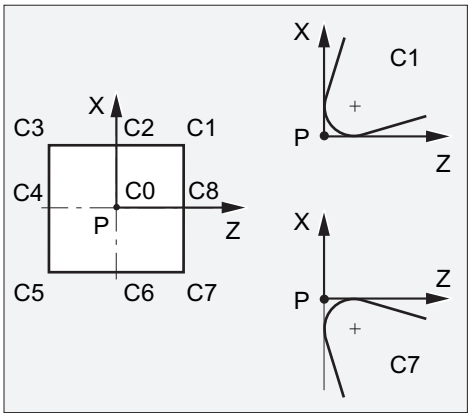



```
%55
N10 G16 R+ G17
N20 T02 D02 M06          Call of tool T02 and correction D02
N30 S180 M40 M03
N40 G00 X100 Y20
N50 Z30                  Taking into account length L.. of D02
N..
N100 D12 Z30             Taking into account length L.. of D12
N..
```

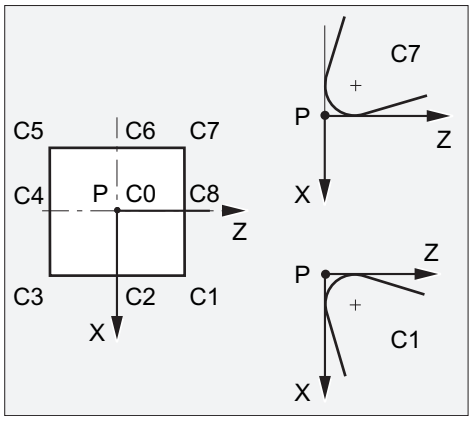
5.11.3.4 Tool tip orientation (C) according to the turret position

The tool tip orientation is defined by codes C0 to C8 (see chapter. 2.2.5).

Rear turret



Front turret



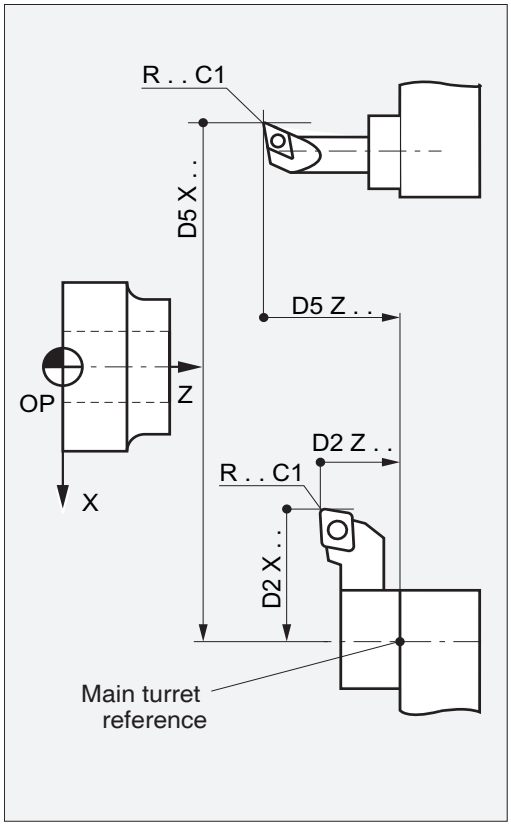
5.11.3.5 Use of tool corrections (X, Z, R, C) on a lathe with two mechanically coupled turrets

Case of a main front turret.

The dimensions of the tools in the secondary turret are defined with respect to the reference point of the main turret.

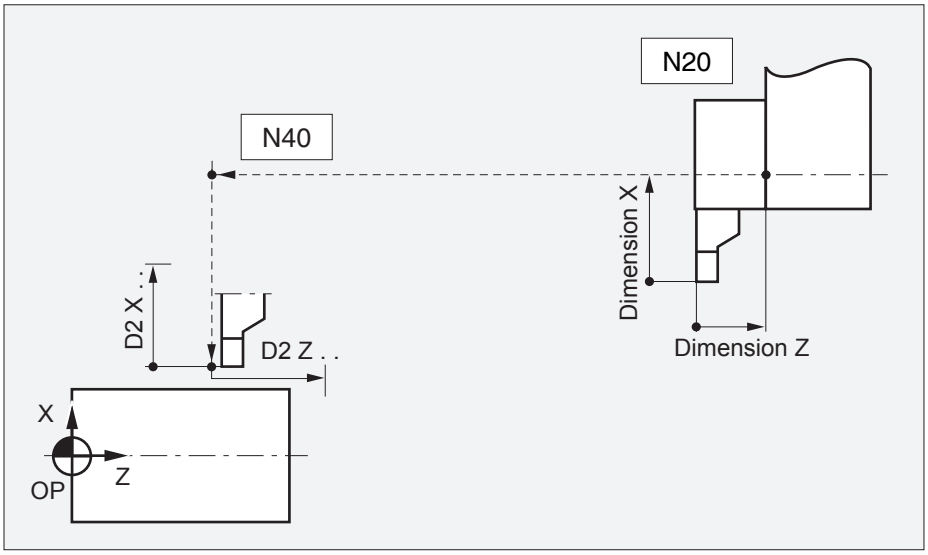
In this case, the tool tip orientation is defined with reference to the front position of the main turret, i.e. C1.

Definition of the tool tip orientation is only important when radius offset G41 and G42 are programmed. With G40 (radius offset cancelled), C0 is defined.



Example

Multiple dimension corrections for a single turning tool

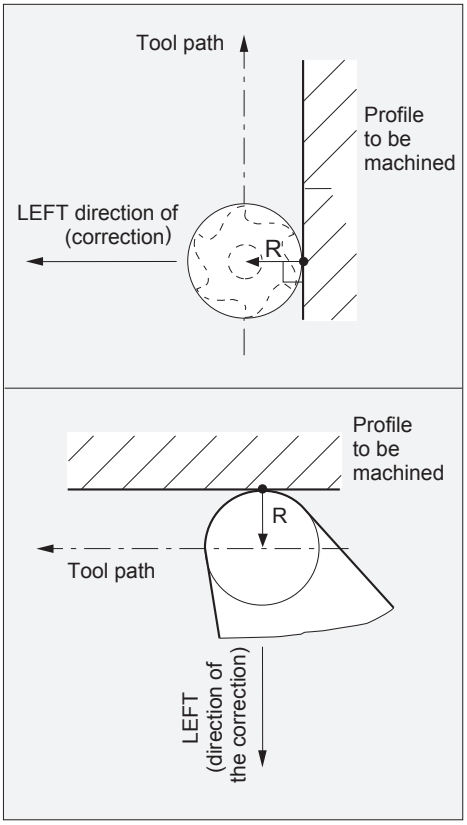



```
%55
N10 G00 G52 X.. Z.. G16 R+
N20 T02 D02 M06                 Call of tool T02 and correction D02
N30 G97 S900 M40 M04
N40 G00 Z100                    Taking into account dimension Z.. of  D02
N50 X60                         Taking into account dimension X.. of  D02
N..
N100 D12 G00 X80 Z100           Taking into account dimensions X.. and
                                Z.. of D12
N..
```

5.11.4 Tool radius compensation ► G41, G42, G40

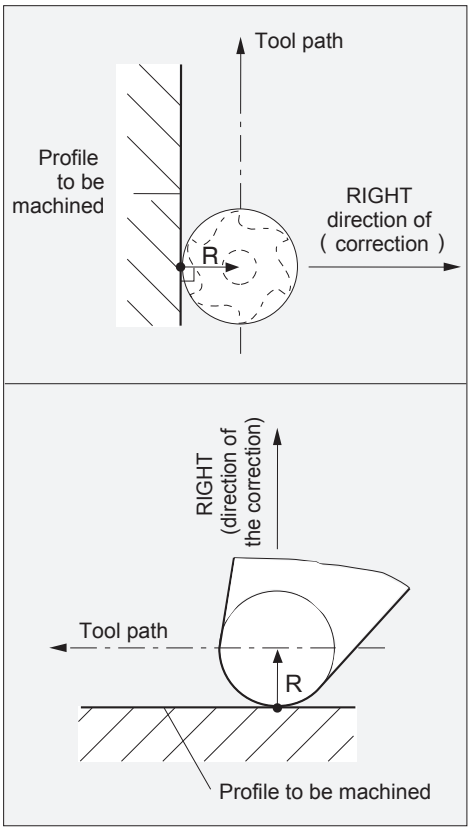
G41	Left radius offset.

The tool paths programmed are corrected (offset to the left) by a value equal to the tool radius (R) declared by corrector D...



G42	Right radius offset.

The tool paths programmed are corrected (offset to the right) by a value equal to the tool radius (R) declared by corrector D...



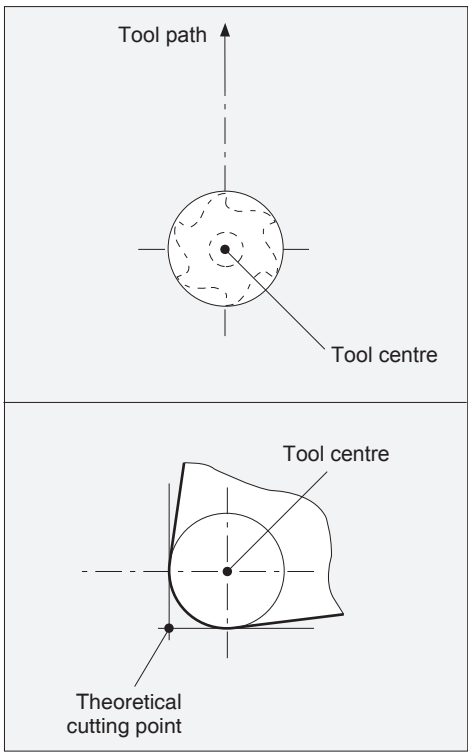
Syntax (XY plane)

N.. [G17] [D..] [G00/G01/G02/G03] **G41/G42** X.. Y..

G17	XY plane selection.
D..	Corrector number containing the tool radius offset.
G00/G01/G02/G03	Linear or circular interpolation.
G41	Radius offset to the left of the profile.
G42	Radius offset to the right of the profile.
X.. Y..	End point.

G40	Radius offset cancel.

The programmed paths are applied to the centre of the tool.



Syntax

N.. [G00/G01] **G40** X.. Y.. Z..

G00/G01	Linear interpolation.
G40	Tool radius offset cancel.
X.. Y.. Z..	End point.

Properties of the Function

Functions G40, G41 and G42 are modal.

Function G40 is the default function.

Cancellation

Functions G41 and G42 cancel one another.

Function G40 cancels functions G41 and G42 as well as function G29 (3D tool correction).

Notes

Functions G41 or G42 allow programming of a part profile in real profile dimensions without taking the tool radius into account.

With radius correction:
- the paths defining the part profile are followed even if the tool radius used and stored is smaller or larger than that of the intended tool,
- the tool is positioned to the left or right of the profile to be machined with respect to the direction of movement along the path.

Radius correction is carried out along a vector perpendicular to the profile with radius R.. declared in correction D as the vector length.

A change of plane (G17/G18/G19) must always be programmed in G40 mode (tool radius offset cancel). For example:

```
N.. ...
N100 G17 G40 X..
N..
N320 G18 G41 X.. Z..
N..
```

Change of plane before calling the radius correction

When changing the direction of correction (change from G41 to G42 or vice versa), it is not necessary to cancel the radius offset by G40.

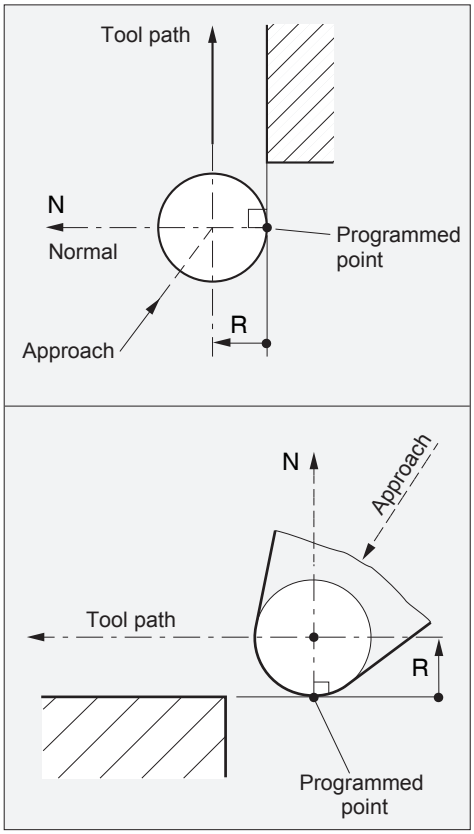
The following functions must be programmed without radius offset (system in state G40), or the system returns error message 140.

- M00 (program stop),
- M01 (optional program stop),
- M02 (end of program),
- G52 (programming with respect to the measurement origin),
- $0 (message transmission),
- L100 to L199 (program variables, see chapter 7.1),
- Parametric programming using E parameters.

5.11.4.1 Tool positioning

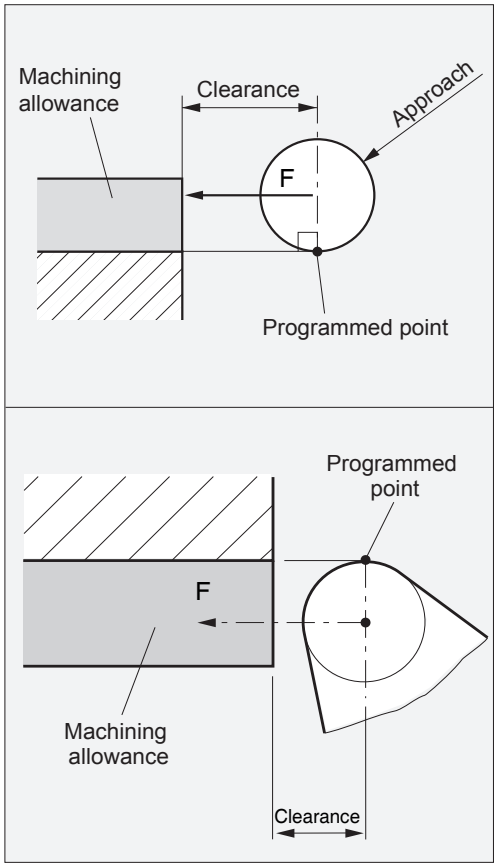
At the end of the first block programmed with radius correction (which must be linear), the tool centre is positioned:

- on the normal (N) to the next path,
- offset from the programmed point by the value of the radius correction (R).



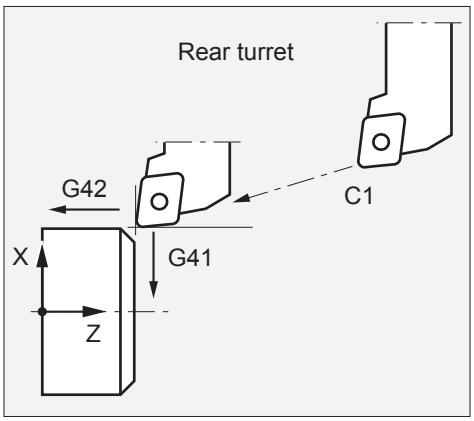
Precaution for positioning the tool

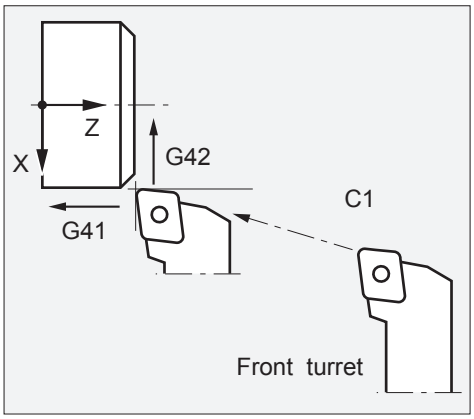
When rapid positioning, provide a clearance with a value higher than the expected tool radius.



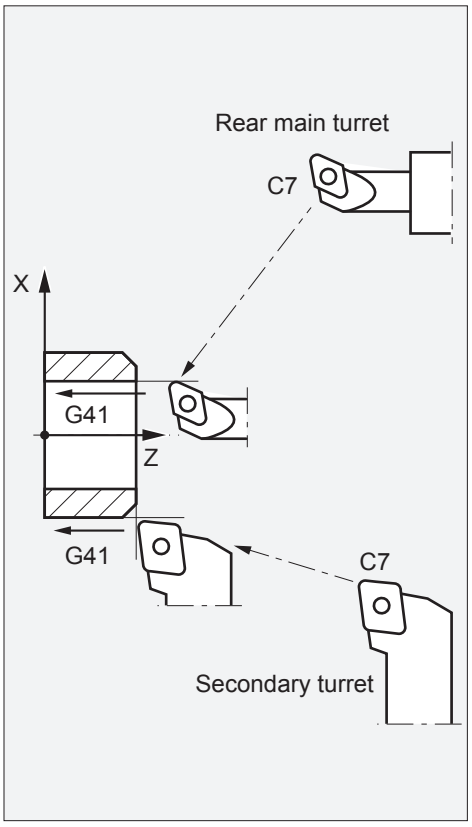
5.11.4.2 Radius correction according to the turret position

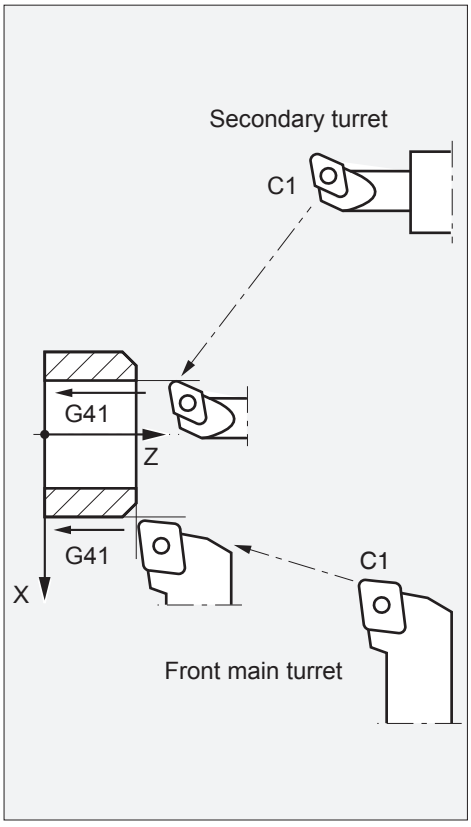
Lathe with one turret





Lathe with two mechanically coupled turrets
The tool tip orientation (C) is defined with reference to the main turret.

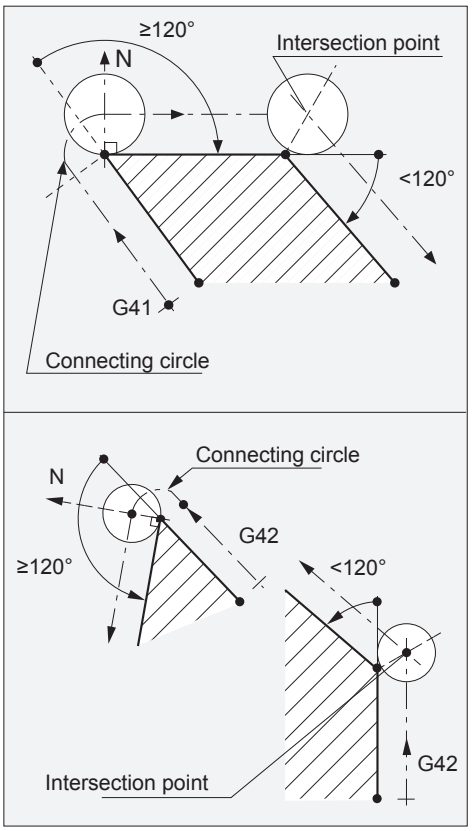


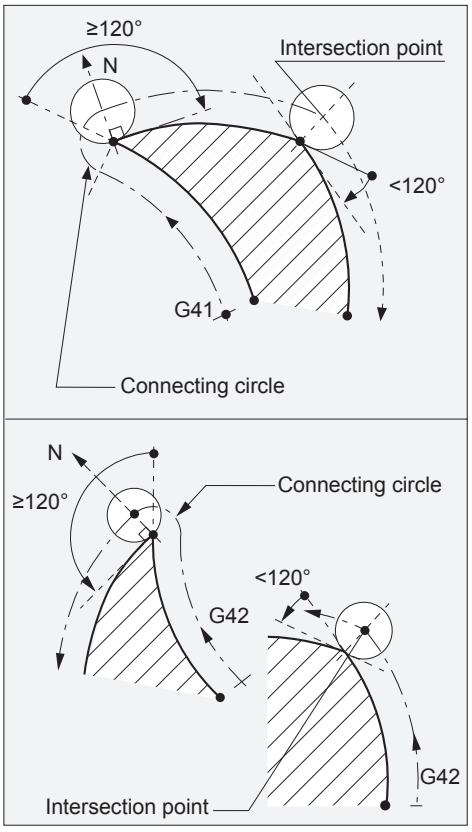


5.11.4.3 Tool outside the profile (line/line or circle/circle)

At the end of the block, the tool centre is positioned:

- offset from the programmed point, on the normal to the next path (angle ≥ 120 degrees) after describing a connecting arc,
- on the point of intersection between the current and next path (angle < 120 degrees).





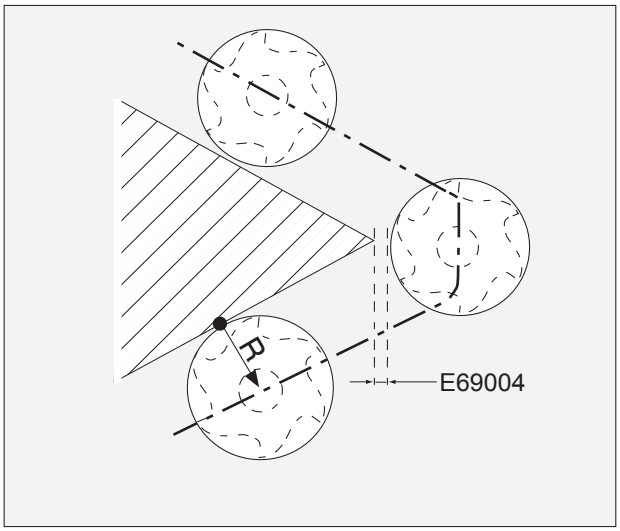
Important remark

With this approach for an angle larger than 120°, the tool stays describes a connection arc while staying in contact with the part which is fine in most cases as it interferes at minima with the rest of the blank.

However in some cases (e.g. Wood working) this cause cause some degradation of the angle. It is therefore possible to define a slight disengagement from the part before the connecting arc.

This disengagement value is programed via parameter E69004 which should be programmed in G40 status. E69004 is valid per axis channel and is automatically limited to 0.4 R.

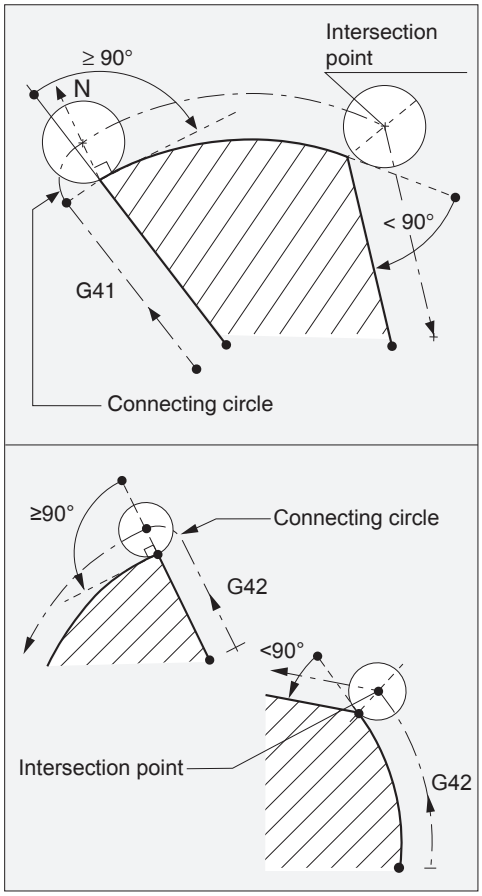
This functionality is only available in the case of Line to Line transition.



5.11.4.4 Tool outside the profile (line/circle)

At the end of the block, the tool centre is positioned:

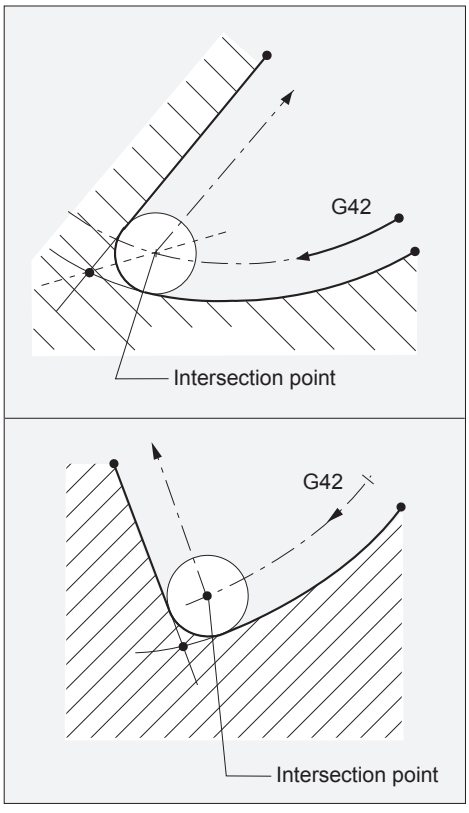
- offset from the programmed point on the normal to the next path (angle ≥ 90 degrees)
  after describing an intersecting arc,
- on the point of intersection between the current and next path with offset (angle < 90 degrees).



5.11.4.5 Tool inside the profile

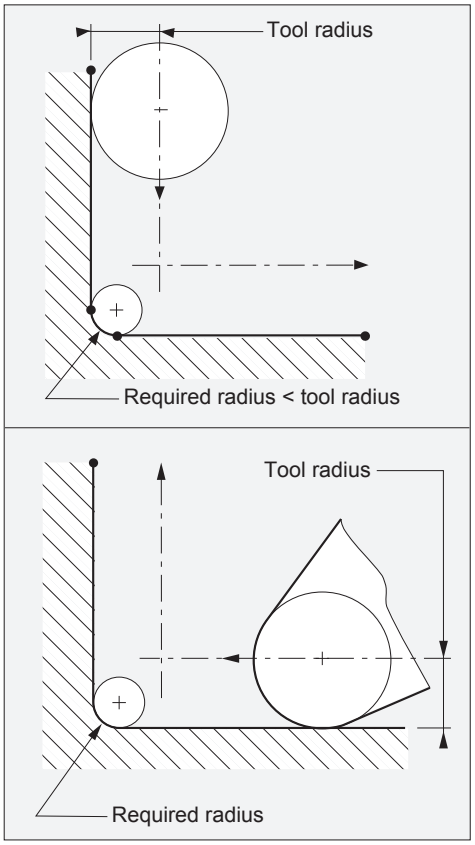
The tool follows the path until its centre reaches the intersection point between the current and next path with offset.

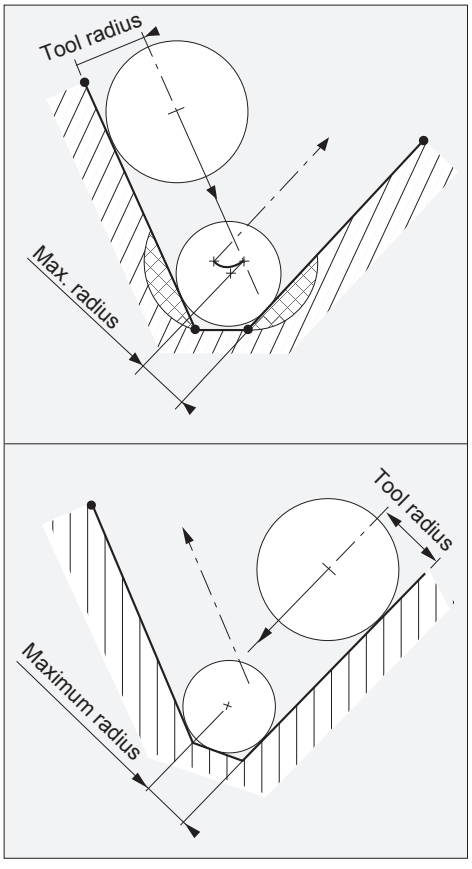
The programmed point is not reached: the radius of the tool leaves a fillet between the two consecutive paths.



5.11.4.6 Tool inside the profile, special cases

When the tool size is too large to be tangent to one of the programmed paths (path radius less than the tool radius or path inaccessible), the system returns error message 149.



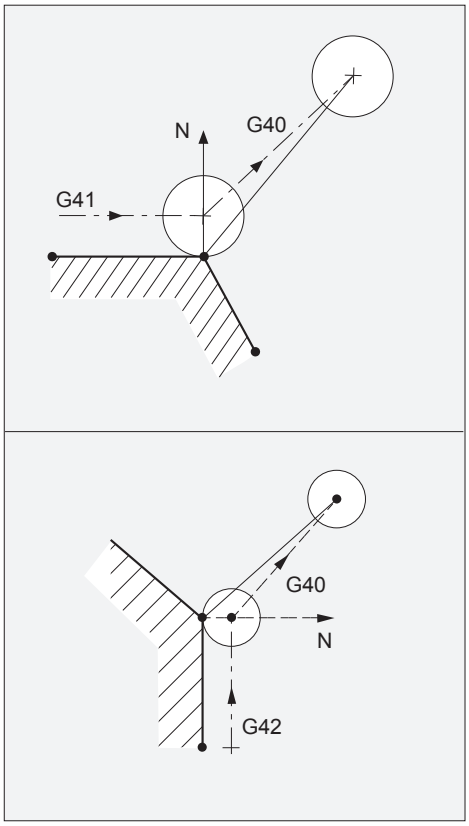


5.11.5 Tool disengagement

At the start of the first block programmed with tool radius offset cancelled, (must be a line), the tool centre starts:

- from the normal to the previous path,
- offset from the programmed point by the value of the radius correction.

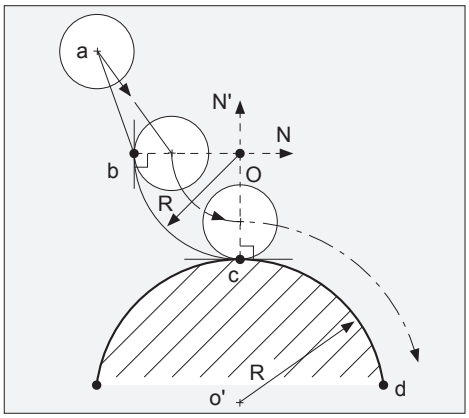
At the end of the move, the tool centre coincides with the programmed point end point.



Examples

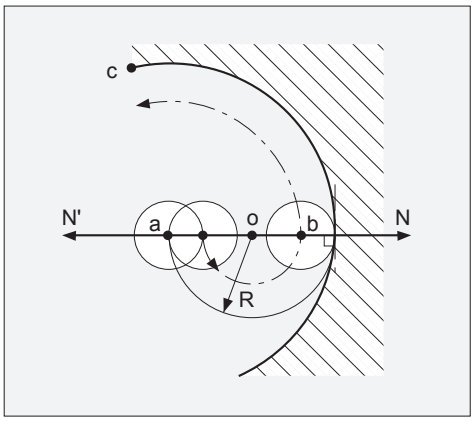
Engagement on an outside circle in G03

```
N.. ...
N.. D04
N40 Xa Ya Za
N50 G01 G41 Xb Yb F..
N60 G03 Xc Yc Io Jo F..
N70 G02 Xd Yd Io' Jo'
N..
```



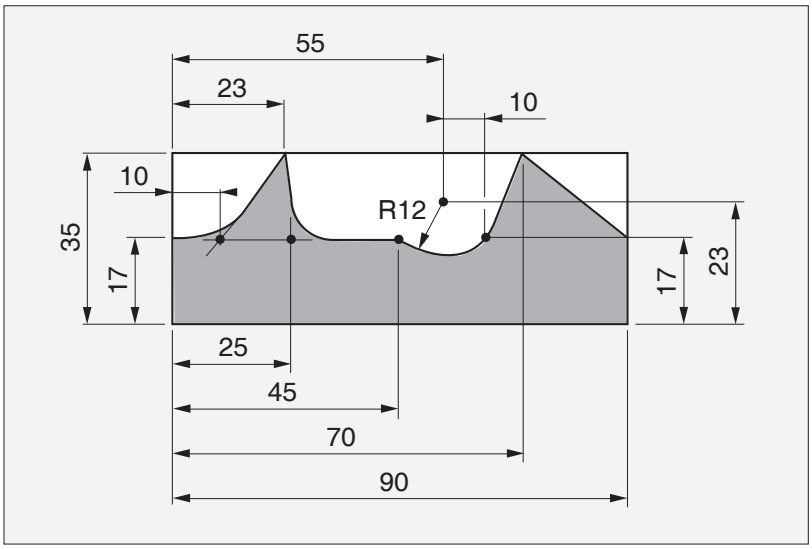
Engagement on an inside circle in G03

```
N.. ...
N.. D05
N50 G41 Xa Ya Za
N60 G03 Xb Yb Io Jo F..
N70 G03 Xc Yc Ia Ja
N..
```

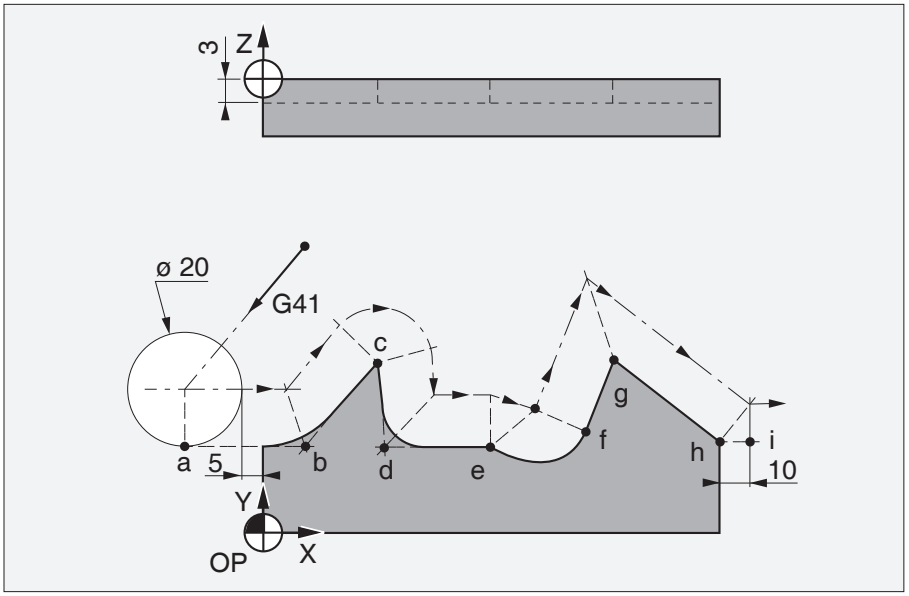


Contouring a profile with radius correction in the XY plane (G17)

Undimensioned radius = Tool radius

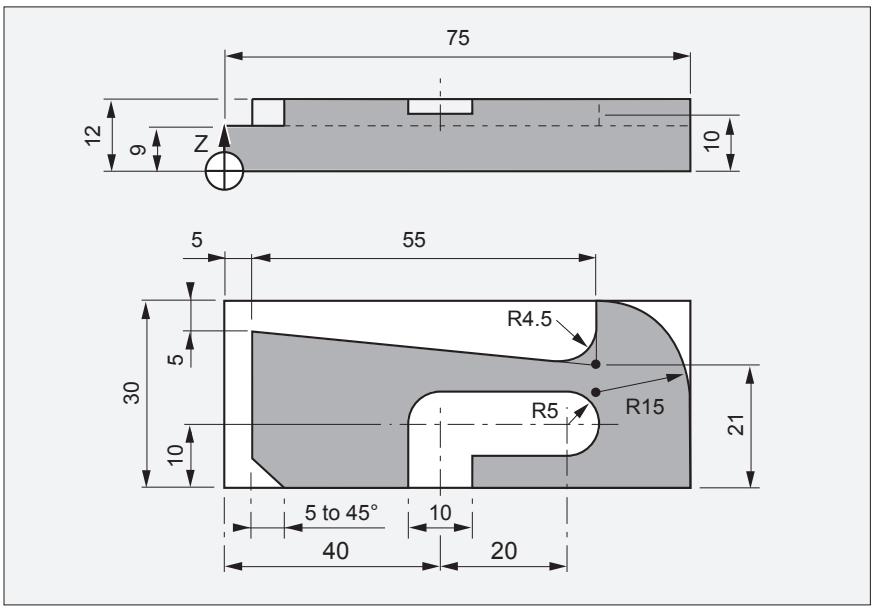


Tool paths (finishing)

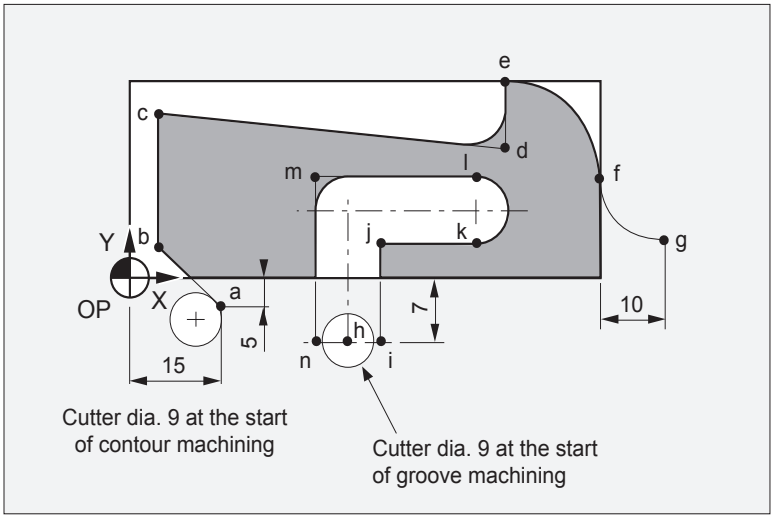



```
%30
N10 G00 G52 Z..
N20 T01 D01 M06 (CUTTER DIAMETER=20)
N30 S300 M03 M40
N40 G00 G41 X-15 Y17                Point a, left radius offset
N50 Z-3                             Tool position in Z
N60 G01 X10 F100 M08                Point b
N70 X23 Y35                         Point c
N80 X25 Y17                         Point d
N90 X45                             Point e
N100 G03 X65 Y17 R12 F50            Point f
N110 G01 X70 Y35 F100               Point g
N120 X90 Y17                        Point h
N130 G40 X100                       Point i, radius offset cancel
N140 G00 G52 Z.. M05 M09            Retraction
N150 M02
```

Contouring a profile and a groove with radius correction in the XY plane (G17), with tool retraction and repositioning



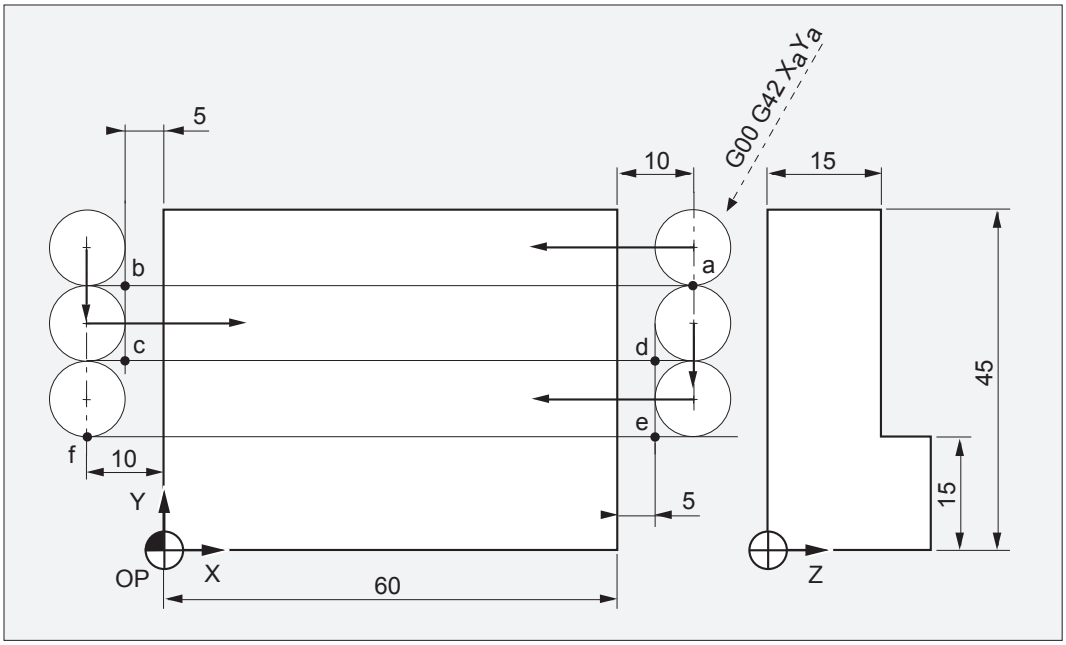
Tool paths




```
%70
N10 G00 G52 Z..
N20 T03 D03 M06 (CUTTER DIAMETER=9)
N30 S700 M40 M03
$0 PROFILE
N40 G00 G41 X15 Y-5          Point a in XY, left radius offset
N50 Z9 M08                   Tool position in Z
N60 G01 X5 Y5 F120           Point b
N70 Y25                      Point c
N80 X60 Y21                  Point d
N90 Y30                      Point e
N100 G02 X75 Y15 R15         Point f
N110 G03 X85 Y5 R10          Point g
N120 G00 G40 Z15 M09         Retraction, radius offset cancel
```

`$0 GROOVEN130 X40 Y-7`	Point h in XY, left radius offset
`N140 Z10 M08`	Tool position in Z
`N150 G41 X45`	Point i
`N160 G01 Y5`	Point j
`N170 X60`	Point k
`N180 G03 X60 Y15 R5 F50`	Point l
`N190 G01 X35 F120`	Point m
`N200 Y-10`	Point n
`N210 G00 G52 Z.. M05 M09`	Retraction
`N220 G40 G52 X.. Y..`	Radius offset cancel
`N230 M02`	

Facing with radius correction in the XY plane (G17) with change from G41 to G42

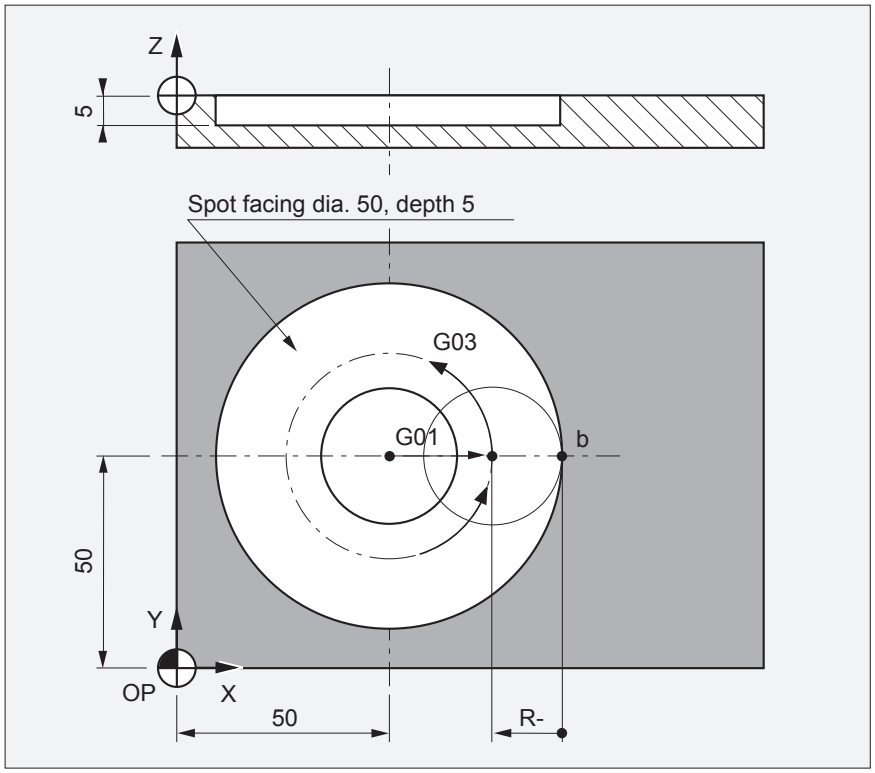



```
%72
N10 G00 G52 Z..
N20 T25 D25 M06
N30 S600 M40 M03
N40 G00 G42 X70 Y35          Point a, right radius offset
N50 Z15                      Tool position in Z
N60 G01 X-5 F200 M08         Point b
N70 Y25                      Point c
N80 G41 X65                  Point d, left radius offset
N90 Y15                      Point e
N100 G42 X-10                Point f, right radius offset
N110 G00 Z250 M09
N120 G40 G52 X.. Y.. M05     Radius offset cancel
N130 M02
```

Spot facing with radius correction applicable to positioning with stop at programmed feed rate (R+/R-)

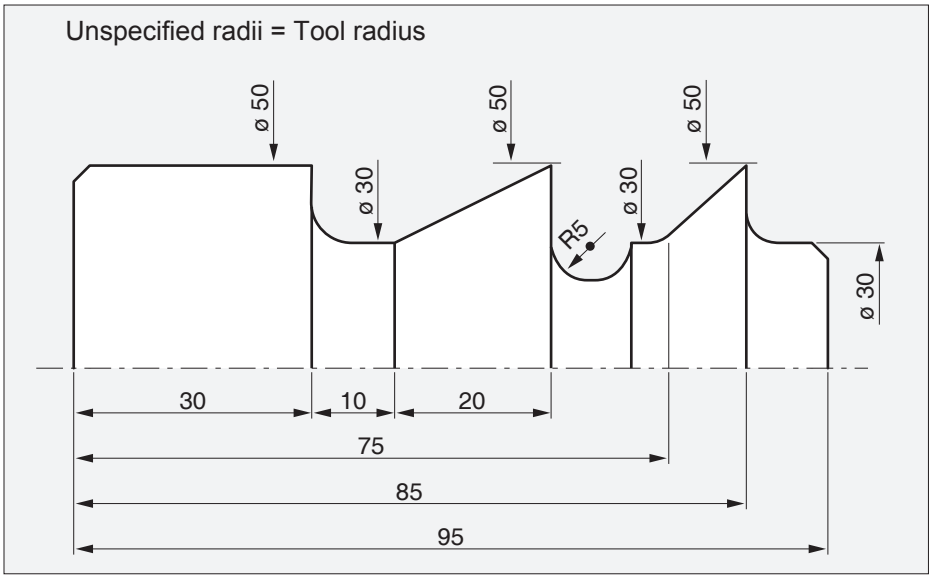
Cutter programmed: diameter = 16.

Cutter declared in correction D12: diameter = 20
(i.e. radius R = 10).

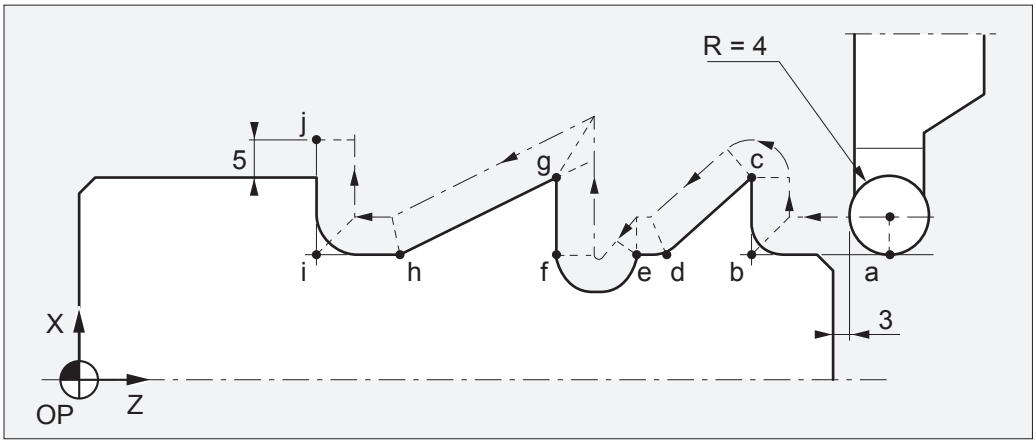



```
%13
N10 G00 G52 Z..
N20 T12 D12 M06 (CUTTER DIAMETER 16)
N30 S300 M40 M03
N40 G00 X50 Y50 Z3                    Point a, approach
N50 G01 Z-5 F50 M08
N60 R- X75 F100                       Stop before programmed point
N70 G41 G03 X75 Y50 I50 J50 F150      Point b
N80 G00 G40 X50
N90 Z50
N..
```

Contouring an external profile with right radius correction

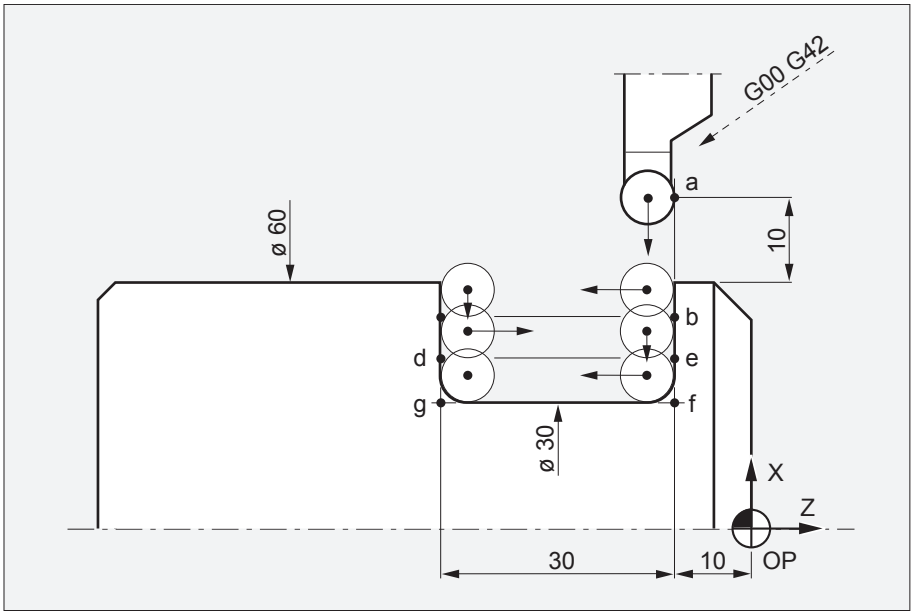


Machining paths (finishing)



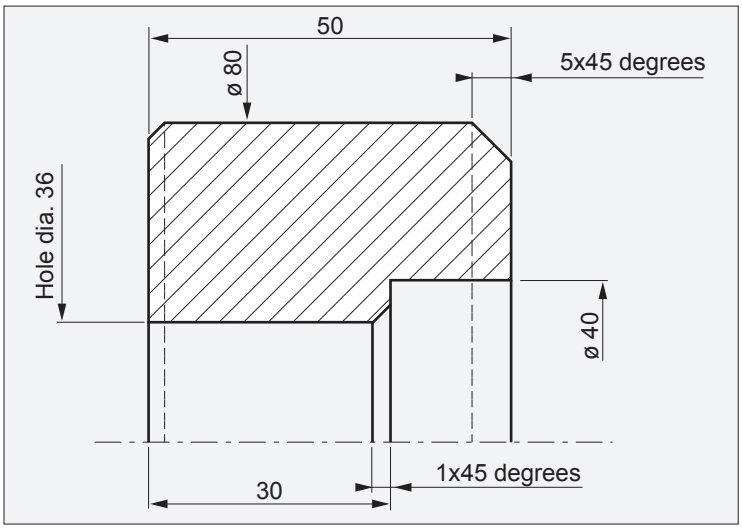

```
%37
N10 G00 G52 X.. Z..                          Tool change position
N20 T03 D03 M06 (COPYING TOOL R=4)
N30 S900 M40 M04
N40 G95 F0.2
N50 G00 X30 Z102                             Point a, right radius offset
N60 G92 S3500
N70 G96 S200
N80 G01 Z85                                  Point b
N90 X50                                      Point c
N100 X30 Z75                                 Point d
N110 Z70                                     Point e
N120 G02 X30 Z60 R5 F0.05                    Point f
N130 G01 X50 F0.2                            Point g
N140 X30 Z40                                 Point h
N150 Z30                                     Point i
N160 X60                                     Point j
N170 G97 S900
N180 G00 G40 G52 X.. Z.. M05                 Radius offset cancel
N190 M02
```

Machining a groove by reciprocating cuts with alternating radius correction (G41-G42)

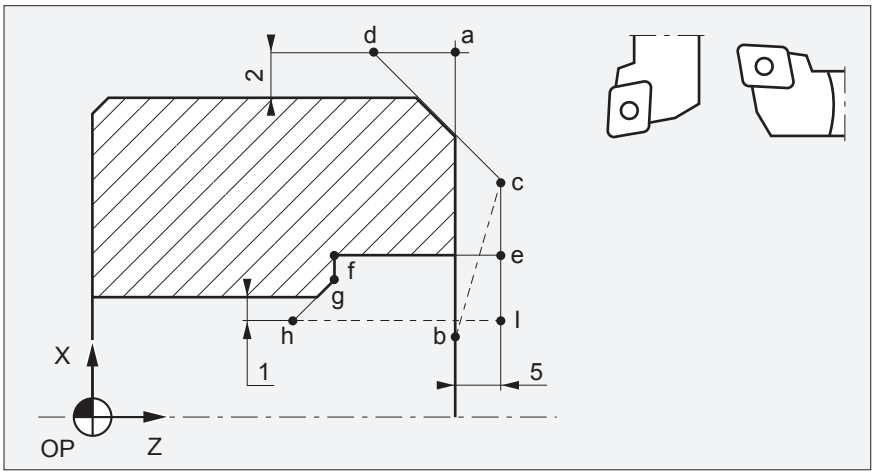



```
%85
N10 G00 G52 X.. Z..                          Tool change position
N20 T01 D01 M06 (COPYING TOOL R=4)
N30 S900 M40 M04
N40 G95 F0.1
N50 G00 G42 X80 Z-10                         Point a, right correction
N60 G92 S3000
N70 G96 S100
N80 G01 X50                                  Point b
N90 Z-40 F0.2                                Point c
N100 G41 X40 F0.1                            Point d, left correction
N110 Z-10                                    Point e
N120 G42 X30                                 Point f, right correction
N130 Z-40 F.2                                Point g
N140 X80
N150 G97 S900
N160 G00 G40 G52 X.. Z.. M05                 Radius offset cancel
N170 M02
```

External machining and internal contouring after tool retraction and positioning



Machining paths




```
%72
N10 G00 G52 X.. Z..                          Tool change position
N20 T01 D01 M06 (OUTSIDE TOOL R=0.8)
N30 S900 M40 M04
N40 G95 F0.12
N50 G00 G41 X84 Z50                          Point a, left correction
N60 G92 S3500
N70 G96 S180
N80 G01 X30                                  Point b
N90 G00 G42 X60 Z55                          Point c, right correction
N100 G01 X84 Z43 F0.08                       Point d
N110 G97 S900
N120 G00 G40 G52 X.. Z..                     Radius offset cancel
N130 T07 D07 M06 (BORING TOOL R=0.4)
N140 G41 X40 Z55                             Point a, left correction
N150 G96 S100
N160 G01 Z30 F0.1                            Point f
N170 X38                                     Point g
N180 X34 Z28                                 Point h
N190 G00 Z55                                 Point i
N200 G97 S900
N210 G40 G52 X.. Z.. M05                     Radius offset cancel
N220 M02
```

5.11.6 Tridimentional tool correction with toroid or spherical tool ► G29

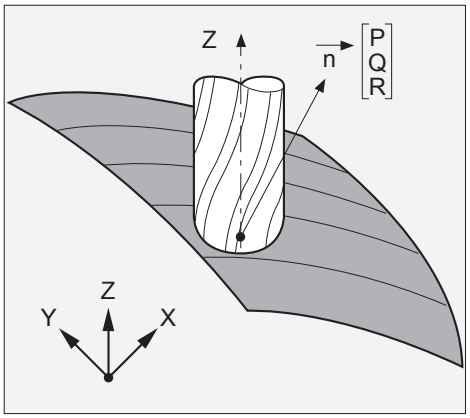
G29	Tridimentional tool correction (3 or 5 axes) with toroid or spherical tool.

Tridimentional tool correction allows machining of tridimentional linear paths taking into account the dimensions of the toroidal or spherical tool used.

3-axis tool correction

For 3-axis correction, the tool axis is parallel to one of the axes of the basic coordinate system defined by the tool axis orientation function G16 ... .

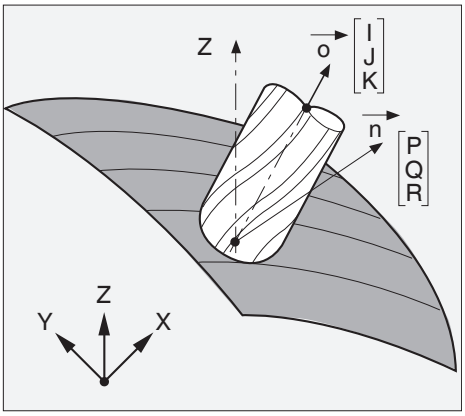
With each programmed point is associated the «material» vector « $\vec{n}$ » normal to the surface to be machined, defined by its P, Q and R coordinates.



5-axis tool correction

For 5-axis correction, the tool axis can be inclined when the machine is equipped with a double twist machining head.

With each programmed point are associated the vector « $\vec{n}$ » normal to the surface to be machined, defined by its P, Q and R coordinates, and the tool orientation vector « $\vec{o}$ », defined by its I, J and K components, plus the twist head angles where required.



Syntax

N.. [D..] [G01] **G29** X.. Y.. Z.. P.. Q.. R.. [I.. J.. K..] [A.. / B.. / C..]

D..	Correction number.
G01	Linear interpolation.
G29	Tridimentional tool correction with toroidal or spherical tool.
X.. Y.. Z..	End point.
P..Q.. R..	Components of the normal vector $\vec{n}$ (material vector) whose origin is defined in the block by the X, Y, Z coordinates of the end point (mandatory in each block).
I.. J.. K..	Components of the tool orientation vector $\vec{o}$ with 5-axis correction (mandatory in each block).
A.. / B.. / C..	Twist head angles with 5-axis correction: A: angle on X, B: angle on Y, C: angle on Z.

Properties of the Function

Function G29 is modal. None of the arguments associated with the function are modal.

Cancellation

Function G29 is cancelled by function G40 or one of functions G41 or G42.

Notes

The presence or absence of vector I J K in a block allows the distinction to be made between 3-axis and 5-axis tool corrections.

For tridimentional correction, the two axes of the basic three-axis system other than the tool axis may be impacted by tool radii R and @, be these axes primary, secondary, carried or independent.

Tridimentional correction can be made on a single point (possibly in MDI manual data input mode).

5.11.6.1 Concept of Surface and Normal Vector

The types of surfaces machined using G29 (irregular surfaces) and the programming requirements (normal vector) mean that tridimentional correction can only be used with machining programs in symbolic language.

Machining is executed by scanning «TABCYL» (word of the APT language defining a tabulated cylinder) by consecutive linear interpolations.

The consecutive program blocks contain the coordinates of the points. The unit normal vector in each point always points outwards from the part.

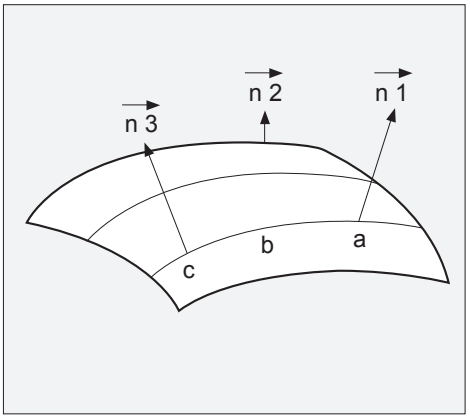
The spacing between the programmed points varies according to the required machining accuracy.

For 3-axis correction, the material vector defined by components P, Q and R must have a length of 100 mm within 1 mm (i.e. +/-0.1%). Otherwise, the system returns error message 145.

The tool correction is applied along the normal vector below:

$$\sqrt{P^2 + Q^2 + R^2} = 1000 \text{ mm}$$

a	X1	b	X2	c	X3
	Y1		Y2		Y3
	Z1		Z2		Z3



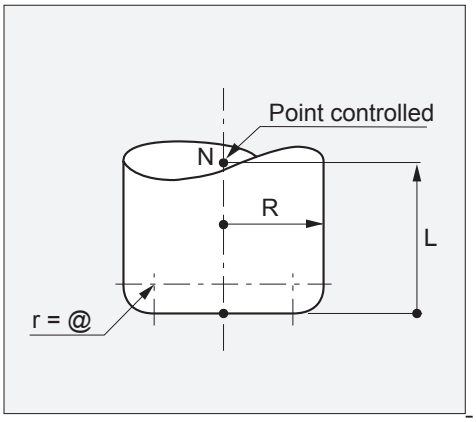
With 5-axis correction, the material vector defined by P, Q and R and the tool direction defined by I, J and K can have any length (the lengths are normalised by the system), but all three components of both vectors must be programmed in each block. Otherwise, the system returns error message 146.

5.11.6.2 Toroidal and Spherical Tools

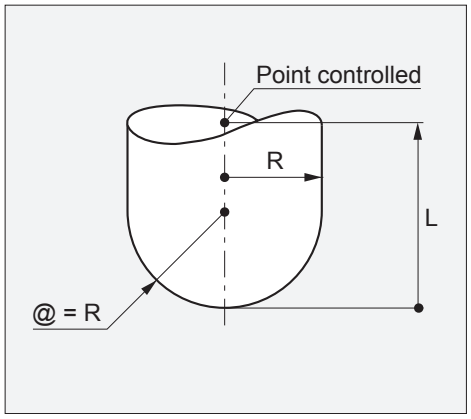
For tridimentional correction, the three corrections to be declared are:
- L.. : tool length,
- R.. : tool radius,
- @..: tool tip radius.

The tool radius «R» must not be confused with the R component of the normal vector.

Toroidal tool



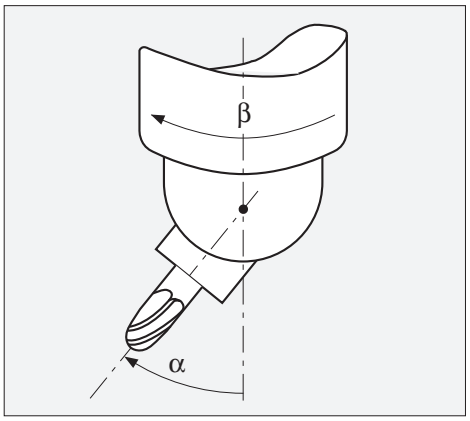
Spherical tool



5.11.6.3 Double Twist Machine Head

Head used with 5-axis correction.

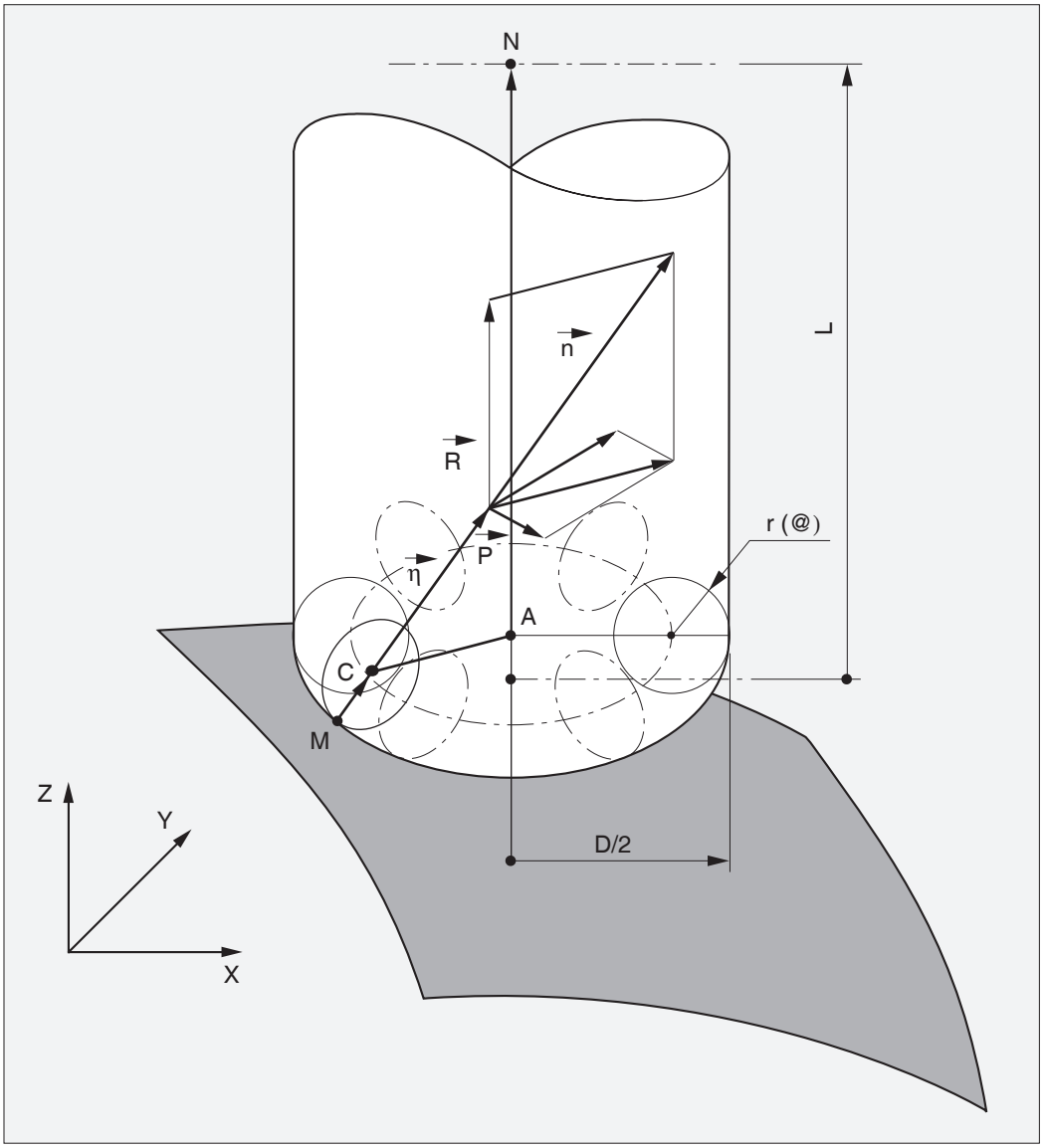
The twist head angles α and β are defined by a pair of rotary axes defined in the NC, i.e.:
- A and B on X and Y,
- A and C on X and Z,
- B and C on Y and Z.



5.11.6.4 Geometric Transformations with 3-Axis Correction

The system computes the tool reference position «N» according to the tangency point «M» and the vector normal to the surface « $\vec{n}$ ».

- $\overrightarrow{MN} = \overrightarrow{MC} + \overrightarrow{CA} + \overrightarrow{AN}$
- $\overrightarrow{MC}$: Vector with length r(@) collinear with vector $\vec{n}$.
- $\overrightarrow{CA}$: Vector with length (D/2 - r) collinear with the projection of vector $\vec{n}$ on the interpolation plane (XY in the figure).
- $\overrightarrow{AN}$: Vector with length (L - r) collinear with the tool axis orientation vector (defined by G16..).



Computation mode

$$\overrightarrow{MC}\begin{vmatrix} \frac{r \cdot P}{\sqrt{P^2+Q^2+R^2}} = r \cdot P \\ \frac{r \cdot Q}{\sqrt{P^2+Q^2+R^2}} = r \cdot Q \\ \frac{r \cdot R}{\sqrt{P^2+Q^2+R^2}} = r \cdot R \end{vmatrix}$$

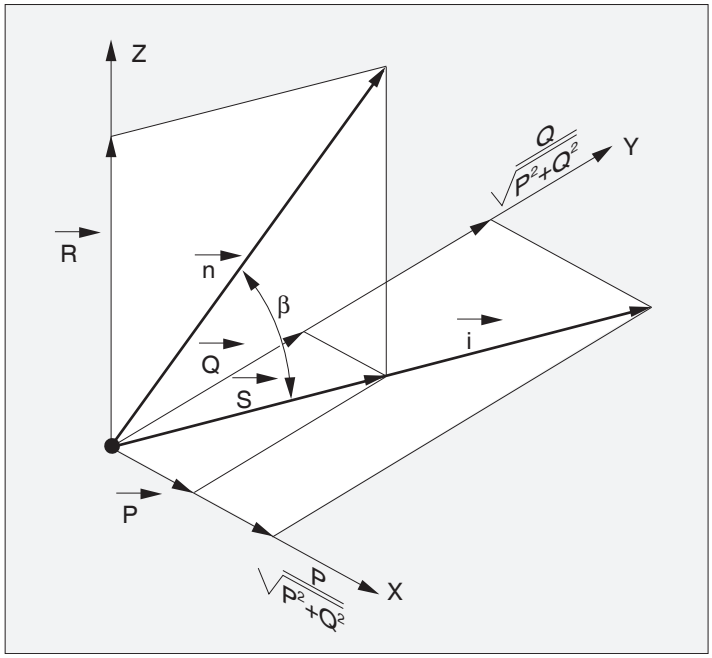
$$XN = XM + \frac{r \cdot P}{1000} + \left(\frac{D}{2} \pm r\right)\frac{P}{\sqrt{P^2+Q^2}}$$

$$YN = YM + \frac{r \cdot Q}{1000} + \left(\frac{D}{2} \pm r\right)\frac{Q}{\sqrt{P^2+Q^2}}$$

$$ZN = ZM + \frac{r \cdot R}{1000} + (L \pm r)$$

$$\overrightarrow{CA}\begin{vmatrix} \left(\frac{D}{2} - r\right)\frac{P}{\sqrt{P^2+Q^2}} \\ \left(\frac{D}{2} - r\right)\frac{Q}{\sqrt{P^2+Q^2}} \\ 0 \end{vmatrix}$$

$$\overrightarrow{AN}\begin{vmatrix} 0 \\ 0 \\ (L - r) \end{vmatrix}$$



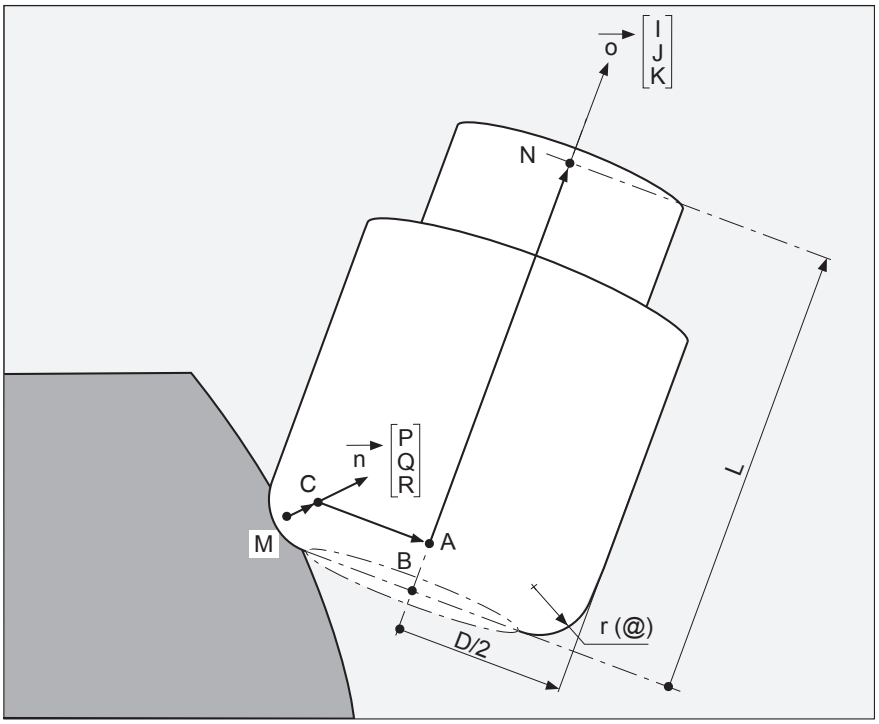
Reminder

In the above equations, P, Q, R are the components of the normal vector with a length of 1000 mm.

5.11.6.5 Geometric Transformations with 5-Axis Correction

The system computes the tool reference position «N» according to the tangency point «M», the vector normal to the surface « $\vec{n}$ » and the tool orientation vector « $\vec{o}$ » (same computation principle as for 3-axis correction).

- M: programmed point
- B: point displayed on the current position coordinate page (AXIS) with respect to OP
- N: point controlled by the system.



Examples

3-axis correction

```
N..
N.. G01 ...
N240 D17
N250 G29 X.. Y.. Z.. P.. Q.. R..
N260 X.. Y.. Z.. P.. Q.. R..
N..
N..
N890 G00 G40 X.. Y..
N..
```

5-axis correction

Twist head axes: A and C.

```
N..
N.. G01 ...
N320 D15
N330 G29 X.. Y.. Z.. P.. Q.. R.. I.. J.. K.. A.. C..
N340 X.. Y.. Z.. P.. Q.. R.. I.. J.. K..
N350 X.. Y.. Z.. P.. Q.. R.. I.. J.. K..
N360 X.. Y.. Z.. P.. Q.. R.. I.. J.. K.. A.. C..
N..
N..
N620 G00 G40 X.. Y..
N..
```

5.11.6.6 Tridimentional Tool Correction with Cylindrical Tool ► G43

G43	Tridimentional tool correction (3 or 5 axes) with cylindrical tool.

Syntax

N.. [D..] [G01] **G43** X.. Y.. Z.. P.. Q.. R.. [I.. J.. K..] [A.. B.. C..]

D..	Correction number.
G01	Linear interpolation.
G43	Tridimentional tool correction with cylindrical tool.
X.. Y.. Z..	Point programmed on the surface.
P.. Q.. R..	Components of the material vector (normalised to 1000) that positions the centre of the tool tip with respect to the programmed point (the offset is obtained by the vector components divided by 1000 and multiplied by the tool radius) (mandatory in each block).
I.. J.. K..	Components of the tool vector normed at 1 by the system giving the tool axis orientation (see Notes).
A.. / B.. / C..	Twist head angles with 5-axis correction: A: angle on X, B: angle on Y, C: angle on Z.

Properties of the Function

Function G43 is modal.

Cancellation

Function G43 is cancelled by function G40.

Notes

With RTCP and twist axes, the tool direction (I J K) does not need to be programmed as it is included in the offset processed by the RTCP function (if this vector is programmed, it is ignored).

Without RTCP and twist axes, if vector I J K is not specified, the tool direction is assumed to be paraxial and is given by function G16.

For information on the RTCP function, see the Flexium Extended Programming Manual M00020 .

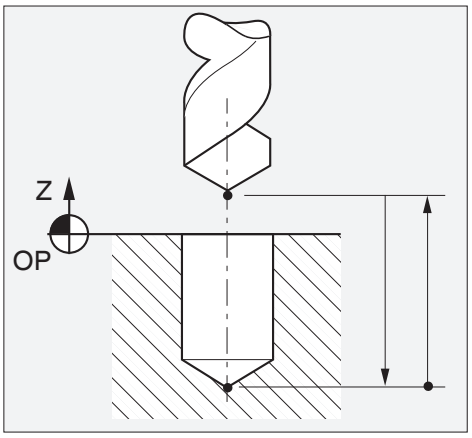
If one of the components of vector PQR is missing or if only one of the components of vector IJK is specified (but not all three), the system returns error message 149.

5.12 Canned Cycles

5.12.1 Cycle Overview ► G8x

G8x (and G31) Canned cycles in the tool axis.

Axes programmable with basic cycles:
- X, Y, Z primary axes,
- U, V, W secondary axes,
- rotary axes A, B or C are used only for positioning.



Syntax (XY plane)

N.. [G17] **G8**x [X.. Y..] Z.. [ER..] [EH..] F..

G17	XY plane selection.
G8x	Canned cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Dimension of the approach (or retraction) plane in the machining axis.
EH..	Dimension of the impact plane in the machining axis.
F..	Feed rate in the cycle.

Properties of the Functions

Functions G8x are modal.

Cancellation

Function G8x is cancelled by function G31, G80 or another function G8x.

Notes

When a cycle (G8x and G31) is programmed, the system must be in state G40 (G41 or G42 tool radius offset cancelled).

The use of program variables L900 to L959 (see chapter 7) is not recommended in a program including canned cycles, since some of these variables could be overwritten when a cycle is called.

Dimensions ER and EH

Dimension ER defines the approach (or retraction) plane on the machining axis. It is assigned to the primary axis (Z) or the secondary axis (W) from the previous block.

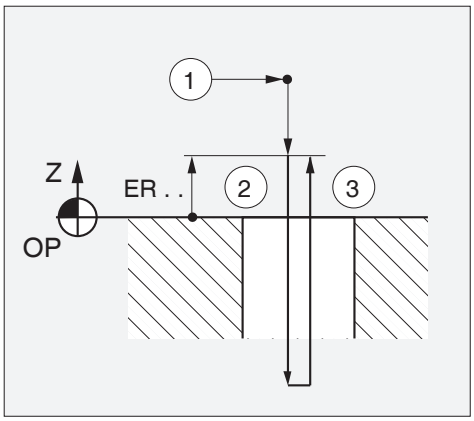
Dimension EH defines the impact plane on the machining axis. It is assigned to the primary axis (Z) or the secondary axis (W) from the previous block.

EH must always be programmed in the same block of the cycle as ER.

Cycle steps with ER (without EH)

- Step 1: Rapid positioning (linear or circular)

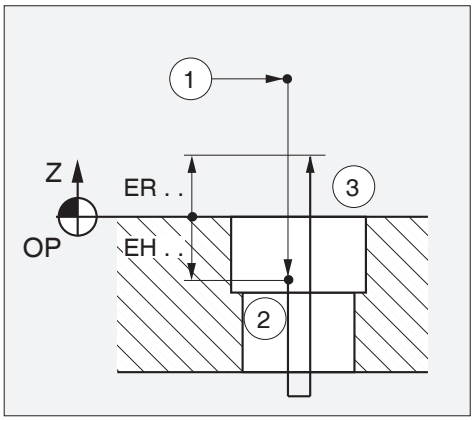
  In the plane then on ER..
- Step 2: Tool penetration to the value programmed on the tool axis (Z).
- Step 3: Retraction along the tool axis (Z) up to ER..



- ER.. not programmed:
  The previous value programmed on the Z axis is used for approach.
- ER.. programmed alone:
  The tool is simply positioned in the Z axis (ie. no cycle).

Cycle steps with EH and ER

- Step 1: Rapid positioning (linear or circular) in the plane, then on EH..
- Step 2: Tool penetration down to the value programmed on the tool axis (Z).
- Step 3: Retraction along the tool axis (Z) up to ER..



- EH and ER.. programmed:

  EH differentiates between the impact plane and the backoff plane.
- EH.. not programmed and ER.. programmed:

  The value of ER is used (ER = EH).

The above descriptions of the cycle steps with ER alone and with ER and EH take into account only the start and end planes. For the detail of these steps, refer to the cycle concerned.

5.12.2 Cancellation of a Canned Cycle ► G80

G80	Canned cycle cancel.

This function is used to cancel canned cycles.

Syntax

N.. **G80**

G80	Canned cycle cancel.

Property of the Function

Function G80 is modal. It is the default function.

Cancellation

Function G80 is cancelled by one of functions G31, G81-G89.

Notes

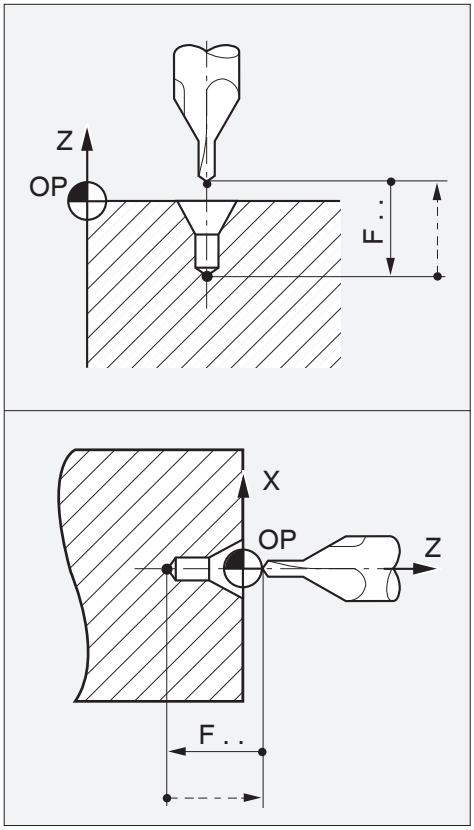
Including function G80 in a cycle subroutine makes the cycle nonmodal.

Example

```
N..
N120 G00 X.. Y.. Z..          Tool positioning
N130 G81 Z-10 F100            Drilling cycle
N140 G80 G00 Z200             Cycle cancel
N..
```

5.12.3 Centre Drilling Cycle ► G81

G81	Centre drilling cycle.



Syntax (XY plane)

N.. [G17] **G81** [X.. Y..] Z.. [ER..] [EH..] [F..]

G17	XY plane selection.
G81	Centre drilling cycle.
X.. Y..	Tool position in the plane.
Z..	End point on a machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
F..	Feed rate in the cycle.

Property of the Function

Function G81 is modal.

Cancellation

Function G81 is cancelled by one of functions G31, G80, G82-G89.

Cycle Steps

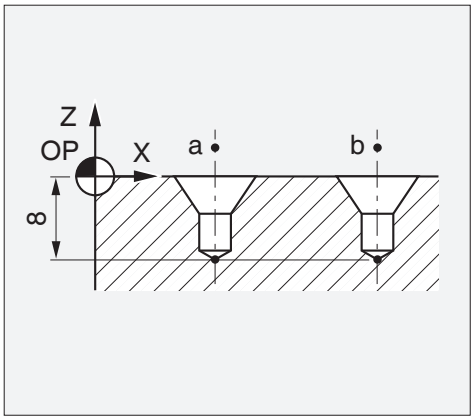
- Step 1: Rapid positioning in the plane.
- Step 2: Infeed at feed rate F..
- Step 3: Rapid retraction in the tool axis.

Example

```
N.. ...
N50 G00 Xa Ya Za
N60 G81 Z-8 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G81 Xa Ya ERa Z-8 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```



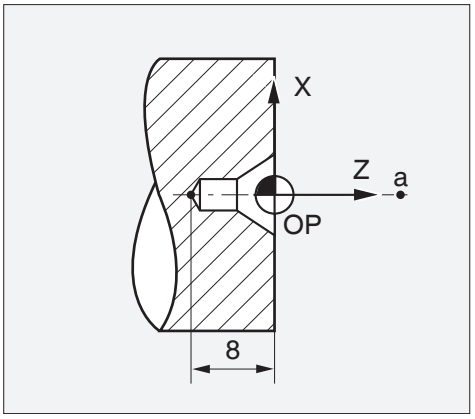
Example

Execution of centre drilling

```
N.. ...
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G81 Z-8
N80 G80 G00 X.. Z..
N..
```

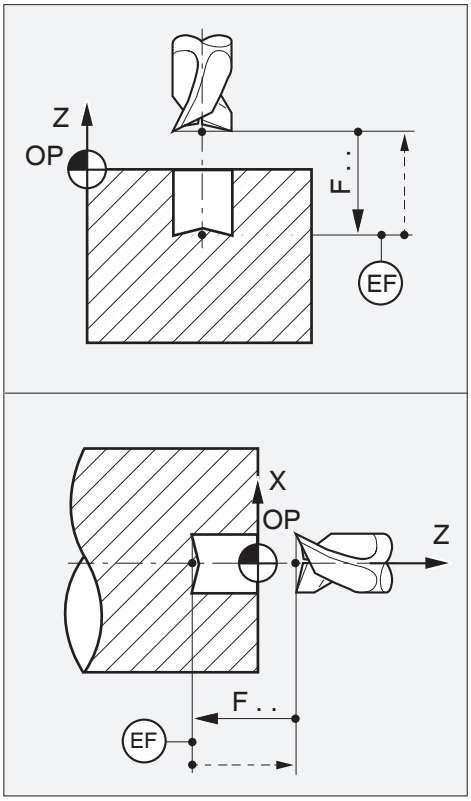
or

```
N..
N50 G94 (or G95) F..
N60 G81 Xa ERa Z-8
N70 G80 G00 X.. Z..
N..
```



5.12.4 Counterboring Cycle ► G82

G82	Counterboring cycle.



Syntax (XY plane)

N.. [G17] **G82** [X.. Y..] Z.. [ER..] [EH..] EF.. [F..]

G17	XY plane selection.
G82	Counterboring cycle.
X.. Y..	Tool position in the plane.
Z..	End point in the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
EF..	Mandatory dwell time in seconds (maximum 99.99 s, format EF022).
F..	Feed rate in the cycle.

Property of the Function

Function G82 is modal.

Cancellation

Function G82 is cancelled by one of functions G31, G80-G81, G83-G89.

Cycle steps

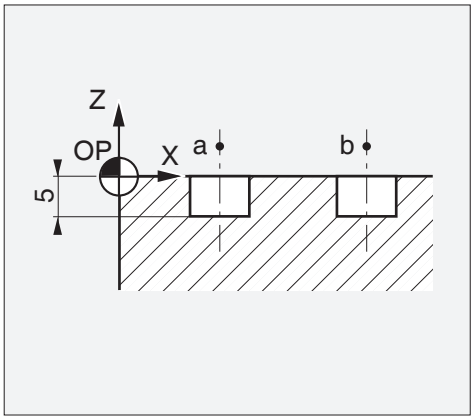
- Step 1: Rapid positioning in the plane.
- Step 2: Infeed at feed rate F..
- Step 3: Dwell at end of drilling or counterboring.
- Step 4: Rapid retraction in the tool axis.

Example

```
N.. ...
N50 G00 Xa Ya Za
N60 G82 Z-5 EF2 F60
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G82 Xa Ya ERa Z-5 EF2 F60
N70 Xb Yb
N80 G80 G00 Z..
N..
```



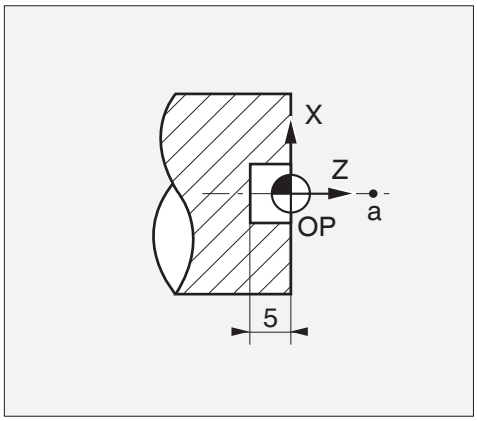
Example

Execution of counterboring

```
N.. ...
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G82 Z-5 EF2
N80 G80 G00 X.. Z..
N..
```

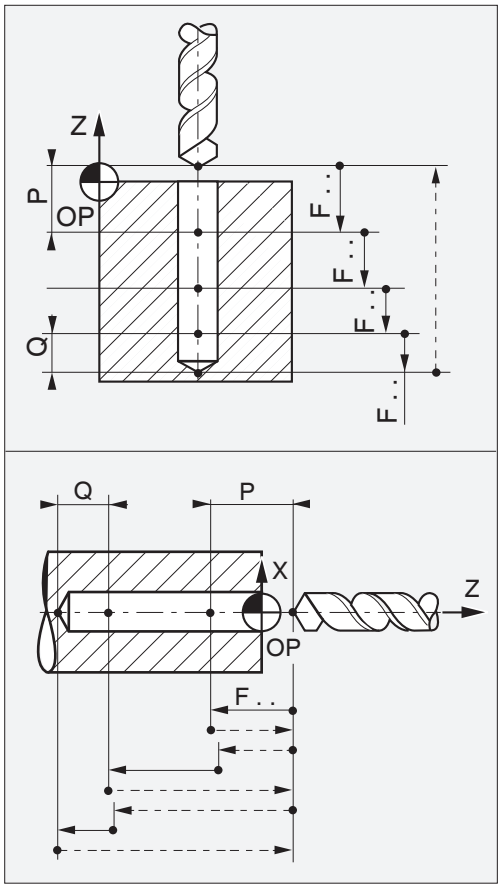
or

```
N..
N50 G94 (or G95) F..
N60 G82 Xa ERa Z-5 EF2
N70 G80 G00 X.. Z..
N..
```



5.12.5 Peck Drilling Cycle ► G83

G83	Peck drilling cycle.



Syntax (XY plane)

N.. [G17] **G83** [X.. Y..] Z.. [ER..] [EH..] [P..] / [ES..] [Q..] [EP..] [F..] [EF..]

G17	XY plane selection.
G83	Peck drilling cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Retraction plane on the machining axis.
EH..	Impact plane in the machining axis.
P..	Value of first peck.
ES..	Number of infeeds (pecks) at constant value (see Fig. 1).
Q..	Value of last peck (optional).
EP..	Backoff after each peck (default 1 mm).
F..	Feed rate in the cycle.
EF..	Dwell at the end of each infeed.

Property of the Function

Function G83 is modal.

Cancellation

Function G83 is cancelled by one of functions G31, G80-G82, G84-G89.

Notes

If addresses P and Q are programmed, the consecutive pecks between P and Q get progressively smaller.

At least one of arguments P and ES must be programmed or the system returns error message 889. If the value of P is greater than delta Z, the system returns error message 881.

Notes related to ES (number of constant infeeds)

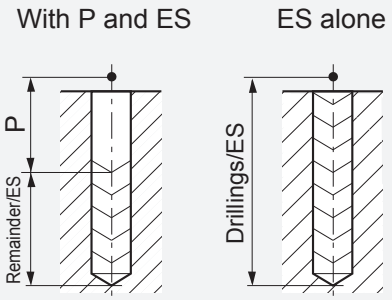
If P and ES are programmed:

The first infeed is equal to P and drilling is continued as a number ES of infeeds.

If ES is programmed alone (without P):

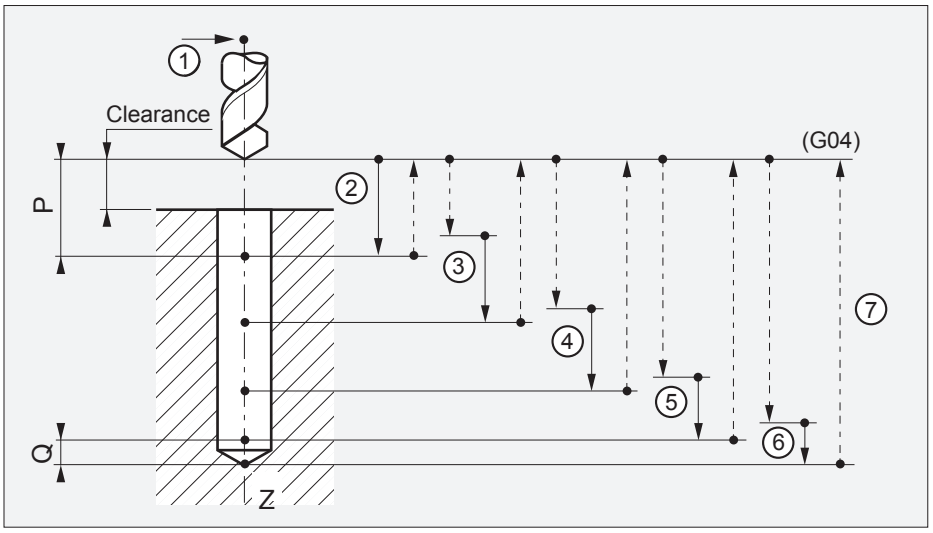
All the drilling is executed as a number ES of infeeds.

Figure 1



Cycle Steps

The steps below are given as an illustration. The number of steps depends on the values programmed in the cycle.



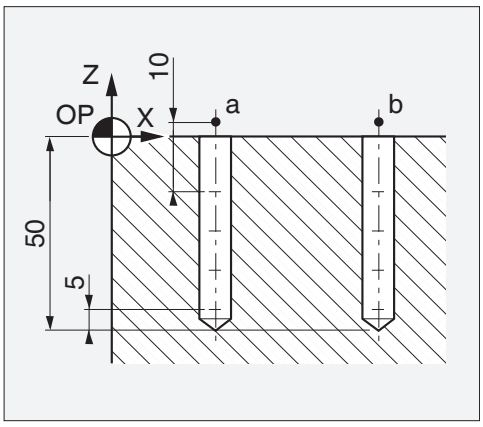
- Step 1: Rapid positioning in the plane.
- Step 2: First peck to depth P.. at feed rate F..
- Rapid retraction to the start point in the tool axis.
- Rapid infeed to 1 mm (or EP..) above depth P..
- Step 3: Second peck at feed rate F..
- Rapid retraction to the start point in the tool axis.
- Rapid infeed to 1 mm (or EP..) above the previous peck.
- Steps 4 and 5: Pecks and retractions same as step 2.
- Step 6: Peck to depth Q.. at feed rate F..
- Step 7: Rapid retraction in the tool axis.
- Possible dwell G04 F.. in the start point.

Example

```
N.. ...
N50 G00 Xa Ya Za
N60 G83 Z-50 P10 Q5 F50
N70 Xb Yb
N80 G80 G00 Z.
N..
```

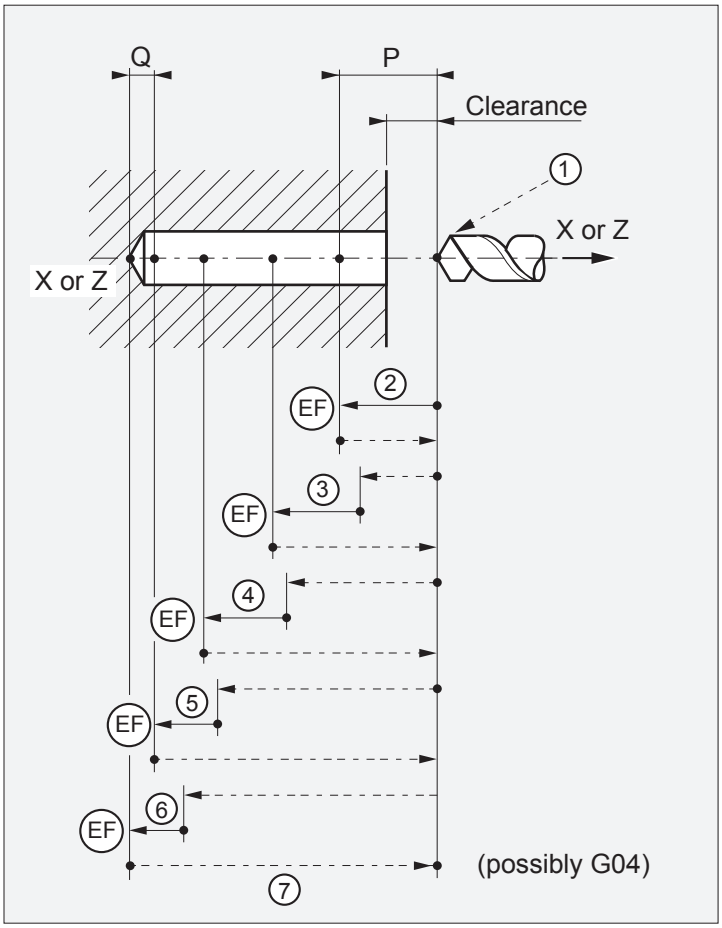
or

```
N.. ...
N60 G83 Xa Ya ERa Z-50 P10 Q5 F50
N70 Xb Yb
N80 G80 G00 Z..
N..
```



Cycle Steps

The steps below are given as an illustration. The number of steps depends on the values programmed in the cycle.

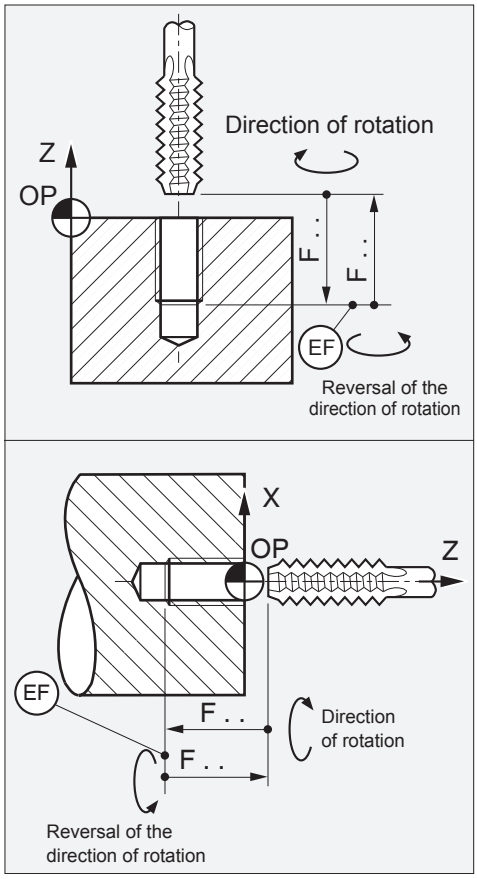


- Step 1: Fast positioning in the machining axis.
- Step 2: First peck to depth P.. in the tool axis at the machining feed rate.
- Possible dwell at the end of the peck.
- Rapid retraction to the start point.
- Rapid infeed to 1 mm (or EP..) above depth P..
- Step 3: Second peck at feed rate.
- Possible dwell at the end of the peck.
- Rapid retraction to the start point.
- Rapid infeed to 1 mm (or EP..) above the previous peck.
- Steps 4 and 5: Pecks and retractions same as step 3.
- Step 6: Peck to depth Q.. at feed rate.
- Step 7: Rapid retraction to the start point.
- Possible dwell G04 F.. in the start point.

5.12.6 Tapping Cycles ► G84

G84	Tapping cycle.

This cycle is used to tap a hole with a floating tap-holder.



Syntax (XY plane)

N.. [G17] **G84** [X.. Y..] Z.. [ER..] [EH..]EF.. [F..]

G17	XY plane selection.
G84	Tapping cycle.
X.. Y..	Tool position in the plane.
Z..	End point in the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
EF..	Dwell time in seconds (maximum 99.99 s, format EF022, default 1 second).
F..	Feed rate in the cycle.

Property of the Function

Function G84 is modal.

Cancellation

Function G84 is cancelled by one of functions G31, G80-G83, G84-G89.

Notes

In this tapping cycle, the feed rate is not coupled to the spindle rotation speed. The tap must be allowed to float to compensate for position errors.

During execution of the cycle, feed rate override by potentiometer is inhibited (value forced to 100%).

Calculation of the feed rate in mm/min

F.. = Tap pitch (in mm) x spindle rotation speed (RPM).

Cycle steps

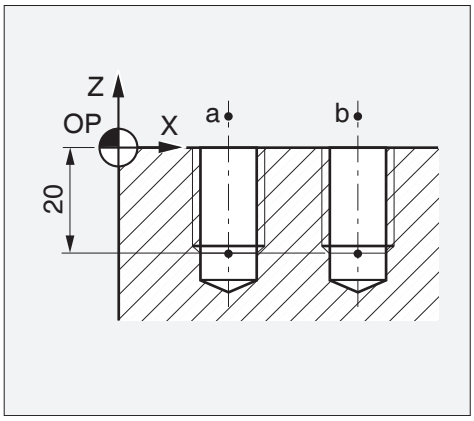
- Step 1: Rapid positioning in the plane.
- Step 2: Infeed at calculated feed rate F..
- Step 3: Reversal of the direction of rotation at depth.
- Step 4: Dwell time at depth.
- Step 5: Retraction at calculated feed rate F.. in the tool axis.

Example

```
N.. ...
N40 S300 M41 M03
N50 G00 Xa Ya Za
N60 G84 Z-20 EF1 F375
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G84 Xa Ya ERa Z-20 EF1 F375
N70 Xb Yb
N80 G80 G00 Z..
N..
```

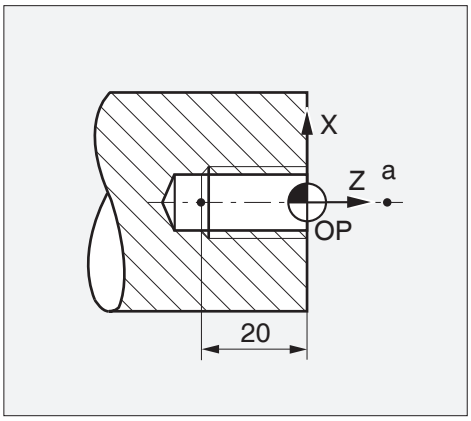


Cycle Steps

- Step 1: Fast positioning in the machining axis.
- Step 2: Infeed at the specified feed rate F..
- Step 3: Reversal of the direction of rotation at the end of tapping.
- Step 4: Dwell at the end of tapping.
- Step 5: Outfeed at the specified feed rate F.. along the tool axis.

Example

Execution of one M8 tapping, pitch 1.25

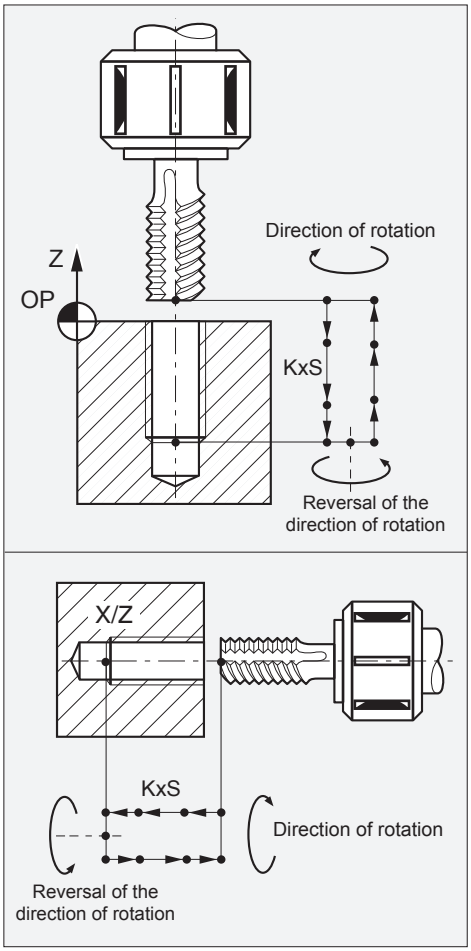



```
N.. ...
N40 S300 M41 M03
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G84 Z-20 EF1
N80 G80 G00 X.. Z..
N..

or

N..
N50 G94 F375
N60 G84 Xa ERa Z-20 EF1
N70 G80 G00 X.. Z..
N..
```

5.12.7 Rigid Tapping Cycle ► G84

G84	Rigid tapping cycle.

With this cycle, the spindle speed controls the tap feed. The feed rate is automatically calculated from the actual spindle speed and (K) programmed pitch.



Syntax (XY plane)

N.. [G17] [M03/M04] [S..] [M40 to M45] **G84** [X.. Y..] Z.. [ER..] [EH..] K.. [EK..]

G17	XY plane selection.
M03/M04	Spindle direction of rotation
S..	Spindle rotation speed.
M40-M45	Spindle ranges.
G84	Tapping cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
K..	Tap pitch in mm (K specifies rigid tapping).
EK..	Spindle outfeed/infeed ratio (the default value of EK = 1).

Property of the Function

Function G84 is modal.

Cancellation

Function G84 is cancelled by one of functions G31, G80-G83, G85-G89.

Notes

When the cycle is called, the tool axis is coupled with spindle rotation.
During execution of the cycle:

- the following error on the tool axis is cancelled during feed at constant speed,
- feed rate and spindle speed override by potentiometer is inhibited (value forced to 100%)

In the end-of-tapping region, the spindle is decelerated then rotation is reversed.
At the end of the cycle, the spindle is returned to its initial direction except if M05 is programed with the cycle, in which case the spindle is stopped at the end when dimension EH is reached.

Rigid tapping can be carried out over several penetrations, but this requires programming several consecutive blocks. With rigid tapping, the system returns error message 899 in the following cases:

- The channel does not control the spindle used or does not provide its measurement.

Tapping Clearance

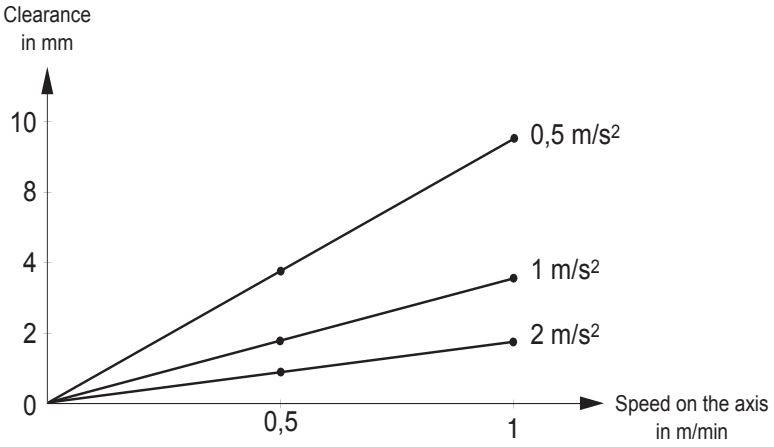
Before starting the cycle, sufficient clearance must be provided to allow the tapping axis to reach correct speed before impacting the material. This clearance depends on the required tapping speed and the acceleration allowed on the axis.

The following chart can be used to approximate the clearance required.

Use of the chart for tapping an M10 hole (pitch = 1.5 mm), for instance:

- Rotation speed = 320 rpm
- Axis feed rate = 480 mm/min or 0.48 m/min
- Acceleration = 0.5 m/s^2.

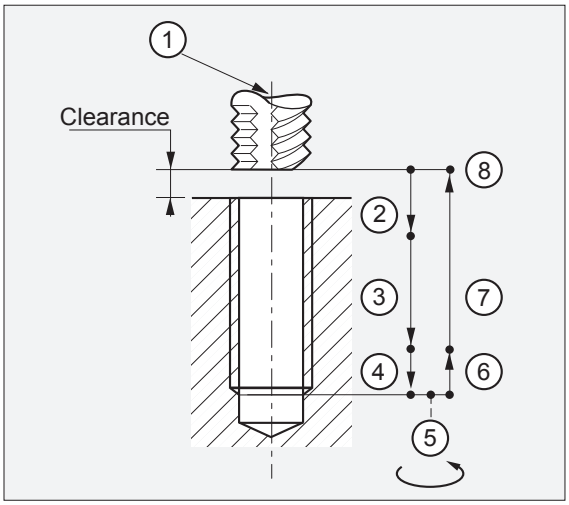
Required clearance read on the chart: approximately 4 mm



It is possible to stop the spindle during rigid tapping. In such case the tapping axis will also stop in synchronism. The cycle will resume if the spindle is restarted.

A Reset is also possible when the spindle is stopped. In such case the cycle will be cancelled. The tap could then be removed via G39+ function (Rigid tap removal).

Cycle Steps

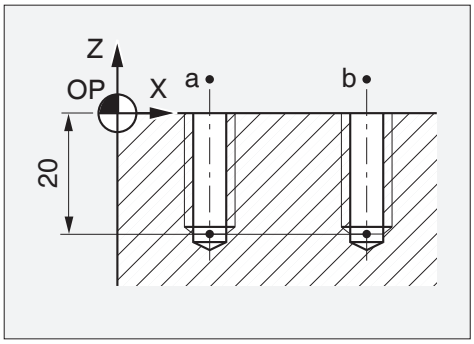


- Step 1: Tool positioning in the hole axis (provide clearance).
- Step 2: Infeed with acceleration of the spindle and feed rate.
- Step 3: Feed at constant speed.
- Step 4: Deceleration before reaching depth.
- Step 5: Reversal of the direction of rotation.
- Step 6: Retraction with acceleration over a distance equal to the deceleration.
- Step 7: Feed at constant speed.
- Step 8: Return of the spindle to the initial direction.

Examples

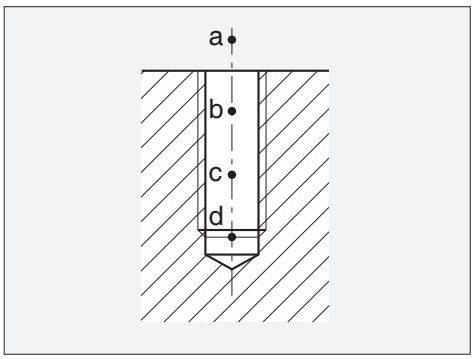
Execution of two rigid tapping operations, pitch 1.50 mm

```
N.. ...
N50 S200 M41 M03
N60 G84 Xa Ya ERa Z-20 K1.5 EK2
N70 Xb Yb
N80 G80 G00 Z..
```

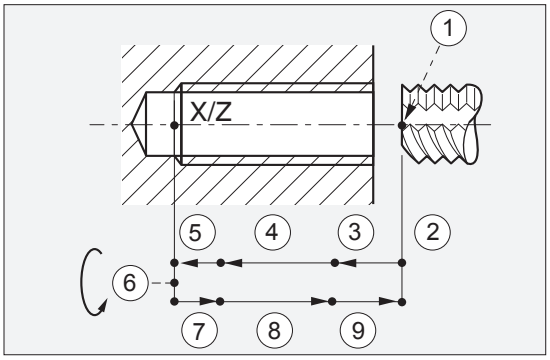


Execution of one rigid tapping operation with several consecutive penetration depths

```
N.. ...
N40 S400 M41 M03
N50 G00 Xa Ya Za
N60 G84 Zb K.. EK..
N70 Zc
N80 Zd
N90 G80 G00 Z..
N..
```



Cycle Steps

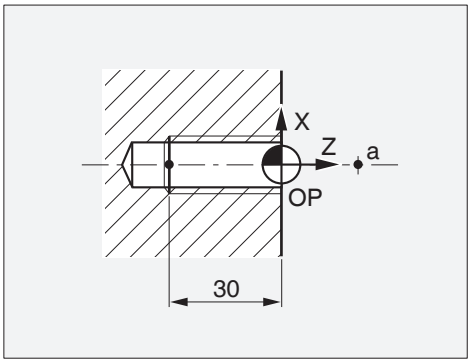


- Step 1: Rapid to start of hole (provide clearance).
- Step 2: Infeed with acceleration of the spindle speed and feed rate.
- Step 3: Feed at constant speed.
- Step 4: Deceleration before reaching depth.
- Step 5: Reversal of the direction of rotation.
- Step 6: Retraction with acceleration over a distance equal to the deceleration.
- Step 7: Feed at constant speed.
- Step 8: Return of the spindle to the initial direction.

Examples

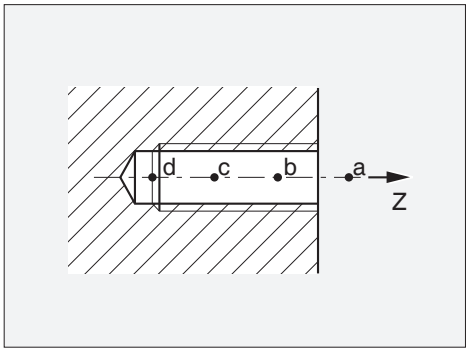
Execution of one M10 hole, pitch 1.50

```
N.. ...
N130 G97 S200 M41 M03
N140 G00 Xa Za
N150 G84 Z-30 K1.5 EK2
N160 G80 G00 G52 X.. Z..
```



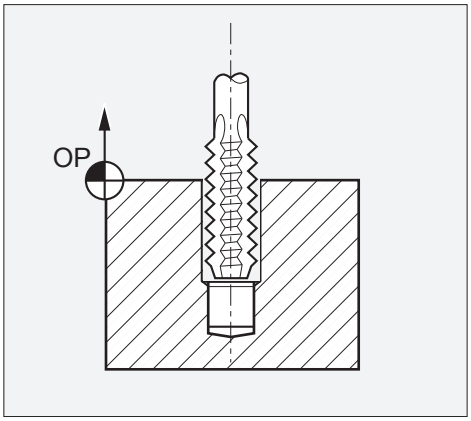
Execution of one tapped hole with several consecutive penetrations

```
N.. ...
N240 G97 S400 M41 M03
N250 G00 Xa Za
N260 G84 Zb K.. EK..
N270 Zc
N280 Zd
N290 G80 G00 G52 X.. Z..
N..
```



5.12.8 Rigid Tap Removal ► G39+

G39+	Rigid Tap Removal.



This function is used if the rigid tapping has been interrupted while it was in progress, then the tool remains caught in the material.

G39+ allows to easily removing the tool from the material, even if homing is not completed and the inclined plane is enabled.

Syntax

[G17] [M64 to M66] **G39+** M3/M4 [M40 to M45] S.. K.. Z..

G17	XY plane.
M64 M66	Functions used if necessary to allow the group to control the spindle.
G39+	Rigid tap removal cycle which calls 4 mandatory parameters and one optional parameter programmed immediately after the function.
M3 or M4	Direction of spindle rotation.
M40 to M45	Speed range (optional).
S..	Spindle speed.
K..	Pitch in mm.
Z..	Axis and signed retraction distance (mm and incremental). Axes U to Z can be programmed if they exist

Notes

The arguments of the cycle can be replaced by E parameters E or L variables.

The functions enabling or inhibiting geometric transformations (inch/metric, scaling factor, DAT, etc.) which are active during a G39+ cycle are processed as follows:

Mirroring and scaling factor are ignored during G39+, then are re-enabled if the program is to be continued (the scaling factor no longer applies to the end point on the tap removal axis).

It should be noted that:

- Cycle G39+ forces use of the spindle in G97
- Since the tool must not be programmed and movement is incremental, the consistency of the tool direction is not checked.

Cycle Steps

- Step 1: Spindle started in the programmed direction.
- Step 2: Tool axis starts to move synchronised with the spindle.
- Step 3: Spindle accelerates up to programmed speed.
- Step 4: When the programmed distance is complete the spindle is stopped and an M02 is automatically generated..

Errors description

Error 1:

- The programmed axis does not exist
- Several axes are programmed or the same axis is programmed several times
- M3/M4 or M40-M45 are programmed twice
- The spindle is not in state M5.

Error 2:

- One of the four mandatory arguments (M3/M4, S, K or Z) is missing
- Another argument was programmed
- An M function other than M3/M4 or M40-M45 -"M02" is programmed
- S is too large (>65536)
- K is negative or zero.

Error 4:

- Rigid tapping option missing.

Error 39:

- S is not compatible with the speed range programmed.

5.12.8.1 Execution of the Rigid Tap Removal Cycle

This cycle can be executed even if homing was not completed on the axis and/or spindle.
The inclined plane can be active. The spindle must be controlled by the group in which the cycle is programmed. At the beginning of the cycle, the spindle is in state M5.

The spindle speed range active at the time of the interrupt should be recovered. If necessary, this speed range can be programmed in G39+.

Functions M3/M4 and M40 to M45 are sent to the PLC, but if subroutines are normally called by these functions, they are not executed during a G39+ cycle. At the beginning of the cycle, the spindle is started in the specified direction at the minimum spindle speed.
Then the spindle is accelerated continuously up to the programmed speed S.

From the beginning of the cycle, movement on the Z axis is linked to the spindle and the following error is cancelled to comply with the specified pitch. The rigid tapping parameters set in P63 are used.

At the end of the cycle:

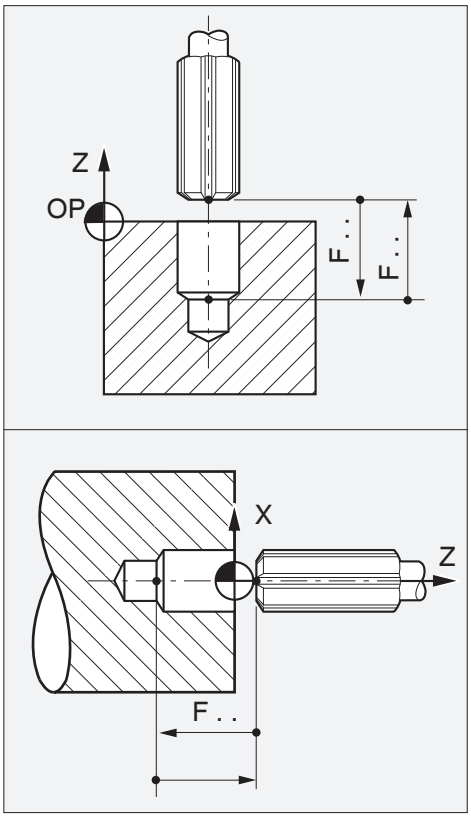
When travel is completed, the spindle is stopped It is recommended to program function M02 as soon as possible.

When travel is completed, the HOMING COMPLETED state is cancelled on the spindle and a new SPINDLE HOMING request is generated.

Movement of the tool axis is enabled during G39+, even if the axis is in HOMING NOT COMPLETED state. However, if the program is to continue, movement is no longer enabled.

5.12.9 Reaming Cycle ► G85

G85	Reaming cycle.



Syntax (XY plane)

N.. [G17] **G85** [X.. Y..] Z.. [ER..] [EH..] [F..] [EF..]

G17	XY plane selection.
G85	Reaming cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
EH..	Impact plane in the machining axis.
ER..	Retraction plane in the machining axis.
F..	Feed rate in the cycle.
EF..	Retraction rate (default = feed rate F..)

Property of the Function

Function G85 is modal.

Cancellation

Function G85 is cancelled by one of functions G31, G80-G84, G86-G89.

Cycle Steps

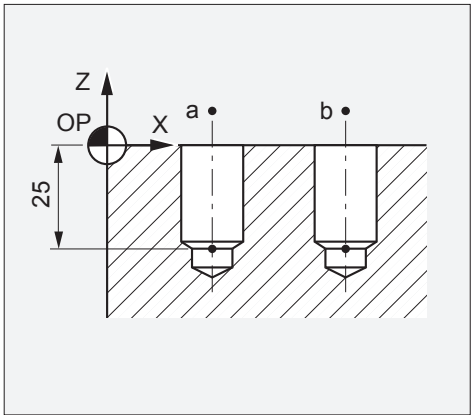
- Step 1: Rapid positioning in the plane.
- Step 2: Infeed at feed rate F..
- Step 3: Retraction at feed rate F.. in the tool axis.

Example

```
N.. ...
N50 G00 Xa Ya Za
N60 G85 Z-25 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G85 Xa Ya ERa Z-25 F80
N70 Xb Yb
N80 G80 G00 Z..
```



Syntax

N.. **G85** X.. / Z.. [ER..] [EH..] [EF..]

G85	Boring cycle.
X.. / Z..	End point on the machining axis.
ER..	Dimension of the retraction plane on the machining axis.
EH..	Impact plane in the machining axis.
EF..	Retraction rate (default = feed rate F..).

Property of the Function

Function G85 is modal.

Cancellation

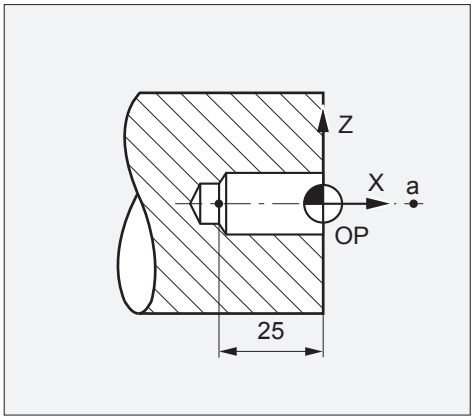
Function G85 is cancelled by one of functions G80-G84, G87 and G89 or functions G64, G65, G66.

Cycle Steps

- Step 1: Fast positioning in the machining axis.
- Step 2: Infeed at feed rate F..
- Step 3: Outfeed at feed rate F.. along the tool axis.

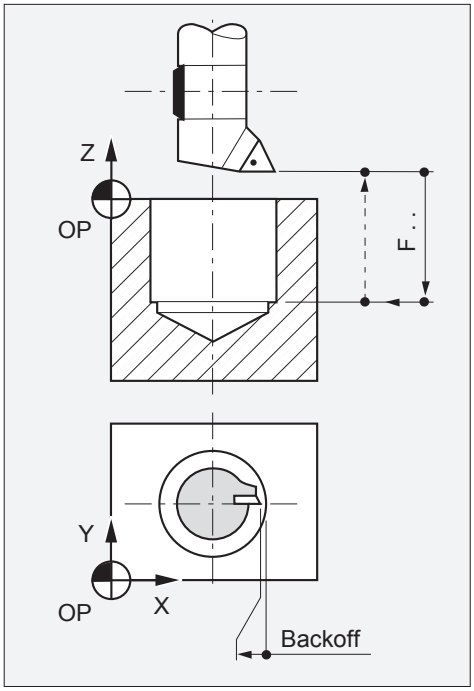
Example

```
N.. ...
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G85 Z-25
N80 G80 G00 X.. Z..
N..

or

N..
N50 G94 (or G95) F..
N60 G85 Xa ERa Z-25
N70 G80 G00 X.. Z..
```



5.12.10 Boring Cycle with Indexed Spindle Stop at the Bottom of the Hole ► G86

G86	Boring cycle with indexed spindle stop at the bottom of the hole.



Syntax (XY plane)

N.. [G17] **G86** [X.. Y..] Z.. [ER..] [EH..] [EC..] [EA..] [EP..] [F..]

G17	XY plane selection.
G86	Boring cycle with indexed spindle stop at the bottom of the hole.
X.. Y..	Tool position in the plane.
Z..	Machining end point.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis
EC..	Value of the indexing position (the default value of EC is the last indexing position programmed).
EA..	Angle between EC.. programmed and the physical angular position of the cutting edge
EP..	Lateral backoff in the hole bottom (default 2 mm)
F..	Feed rate in the cycle.

Property of the Function

Function G86 is modal.

Cancellation

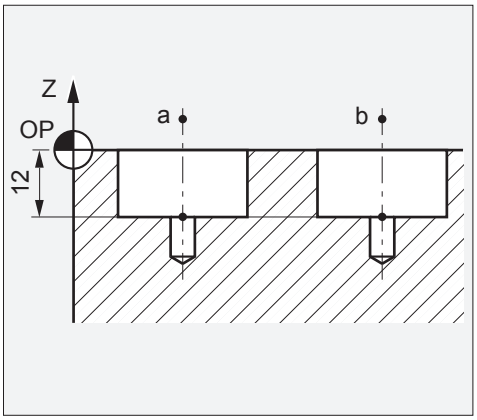
Function G86 is cancelled by one of functions G31, G80-G85, G87-G89.

Cycle Steps

- Step 1: Rapid positioning in the plane.
- Step 2: Infeed at feed rate F..
- Step 3: Indexed spindle stop at depth.
- Step 4: Lateral backoff 2 mm (or EP..) on the indexing axis.
- Step 5: Rapid retraction in the tool axis.

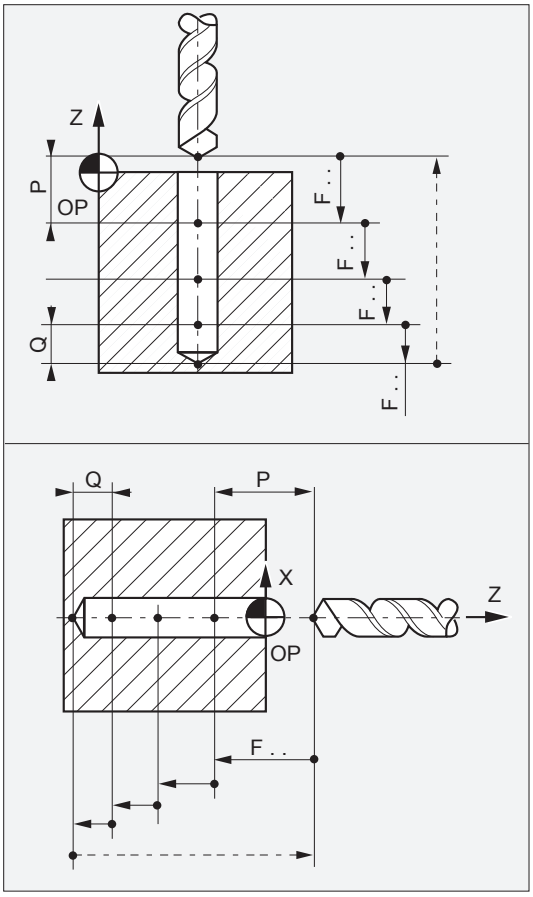
Example

```
N.. ..
N50 G00 X.. Y.. Za
N60 G86 Xa Ya Z-10 EC.. F30
N70 Xb Yb
N80 G80 G00 Z..
N..
```



5.12.11 Drilling Cycle with Chip Breaking ► G87

G87	Drilling cycle with chip breaking.



Syntax (XY plane)

N.. [G17] **G87** [X.. Y..] Z.. [ER..] [EH..] [P..] / [ES..] [Q..] [EP..] [EF..] [F..]

G17	XY plane selection.
G87	Drilling cycle with chip breaking.
X.. Y..	Tool position in the plane.
Z..	End point in the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
P..	Value of first infeed.
ES..	Number of infeeds at constant value (see Fig. 1)
Q..	Value of last infeed (optional).
EP..	Backoff between two infeeds (default no backoff, EP = 0).
EF..	Mandatory dwell time in seconds.
F..	Feed rate in the cycle.

Property of the Function

Function G87 is modal.

Cancellation

Function G87 is cancelled by one of functions G31, G80-G86, G88-G89.

Notes

If addresses P and Q are programmed, the consecutive infeeds between P and Q get progressively smaller. At least one of arguments P and ES must be programmed or the system returns error message 889. If the value of P is greater than delta Z, the system returns error message 881.

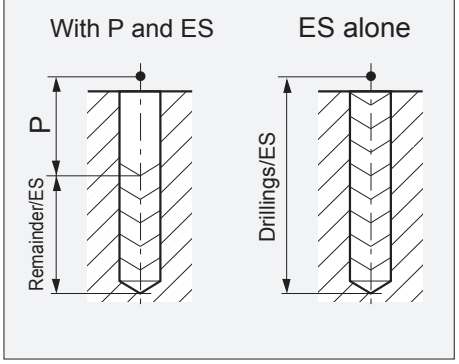
Notes related to ES (number of constant infeeds)

- If P and ES are programmed:

  The first infeed is equal to P and drilling is continued as a number ES of infeeds.
- If ES is programmed alone (without P):

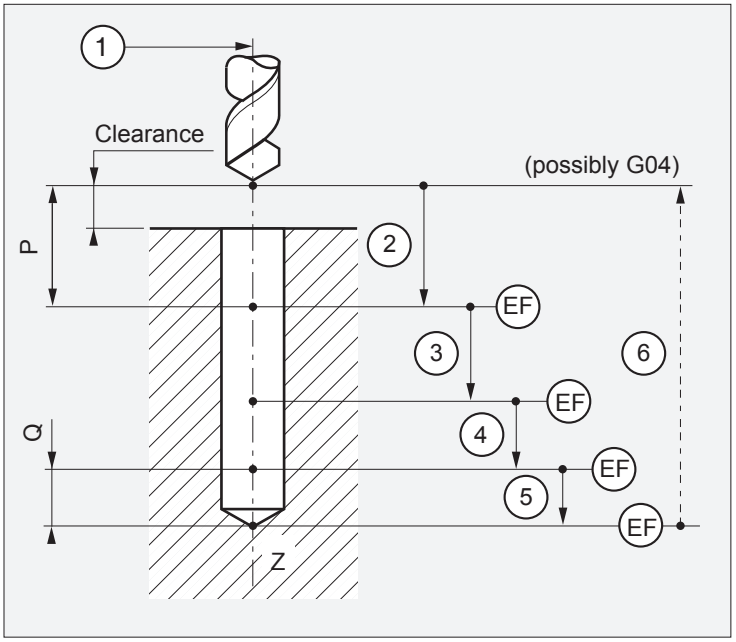
  All the drilling is executed as a number ES of infeeds.

Fig.1



Cycle Steps

The steps below are given for reference. The number of steps depends on the values programmed with the cycle.



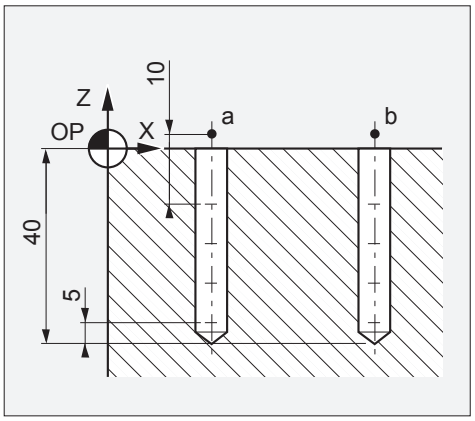
- Step 1: Rapid positioning in the plane.
- Step 2: Infeed to depth P.. at feed rate F..
- Dwell at the end of each infeed (possible backoff by value EP..).
- Steps 3 and 4: Successive infeeds and dwells (with possible backoffs by value EP..) same as step 2.
- Step 5: Infeed to depth Q.. at feed rate F..
- Dwell at depth.
- Step 6: Rapid retraction in the tool axis.
- Possible dwell G04 F.. at the start point.

Example

```
N.. ...
N50 G00 Xa Ya Za
N60 G87 Z-40 P10 Q5 EF1 F40
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G87 Xa Ya ERa Z-40 P10 Q5 EF1 F40
N70 Xb Yb
N80 G80 G00 Z..
N..
```



Syntax

N.. **G87** X.. / Z.. [ER..] [EH..] [P..] / [ES..] [Q..] [EP..] [EF..]

G87	Drilling cycle with chip breaking.
X.. / Z..	End point on the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis
P..	Value of first infeed.
ES..	Number of infeeds at constant value (see Fig. 1)
Q..	Value of last infeed.
EP..	Backoff between two infeeds (default no backoff, EP = 0).
EF..	Dwell at the end of each infeed.

Property of the Function

Function G87 is modal.

Cancellation

Function G87 is cancelled by one of functions G80-G85 and G89 or functions G64, G65, G66.

Notes

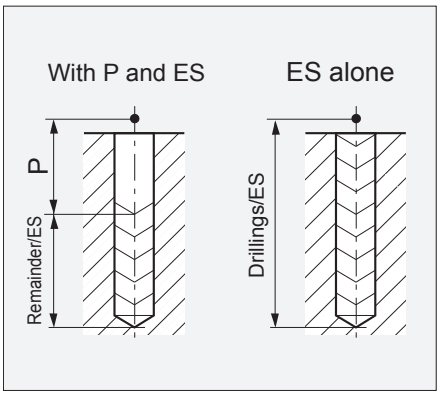
If addresses P and Q are programmed, the consecutive infeeds between P and Q get progressively smaller. At least one of arguments P and ES must be programmed or the system returns error message 889. If the value of P is greater than delta Z, the system returns error message 881.

Notes related to ES (number of constant infeeds)

- If P and ES are programmed:

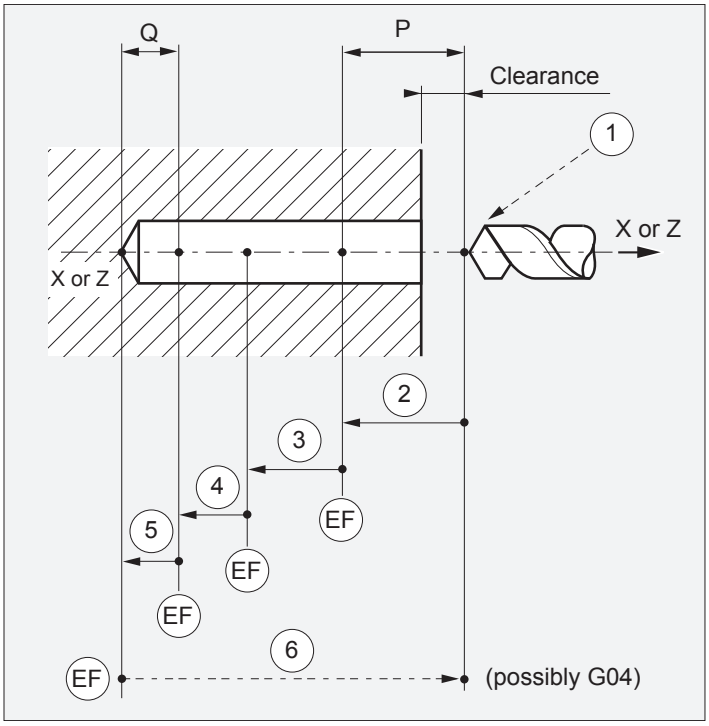
  The first infeed is equal to P and drilling is continued as a number ES of infeeds.
- If ES is programmed alone (without P):

  All the drilling is executed as a number ES of infeeds.



Cycle Steps

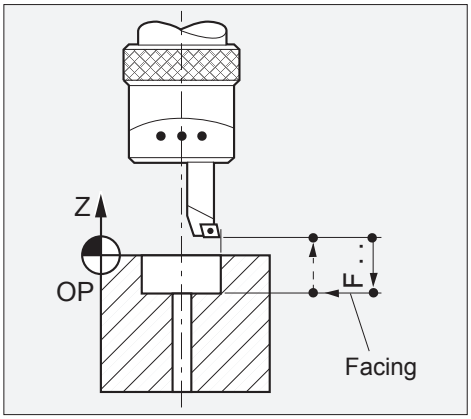
The steps below are given as an illustration. The number of steps depends on the values programmed with the cycle.



- Step 1: Fast positioning in the machining axis.
- Step 2: First infeed of depth P.. at programmed feed rate.
- Possible dwell at the end of the infeed.
- Steps 3 and 4: Consecutive infeeds and dwells as step 2.
- Step 5: Infeed depth Q.. at the feed rate.
- Possible dwell at the end of the peck.
- Step 6: Rapid retraction in the tool axis.
- Possible dwell G04 F.. at the start point.

5.12.12 Boring and Facing Cycle ► G88

G88	Boring and facing cycle.



Syntax (XY plane)

N.. [G17] **G88** [X.. Y..] Z.. [ER..] [EH..] [F..]

G17	XY plane selection.
G88	Boring and facing cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
F..	Feed rate in the cycle.

Property of the Function

Function G88 is modal.

Cancellation

Function G88 is cancelled by one of functions G31, G80-G87, G89.

Cycle Steps

- Step 1: Rapid positioning in the plane.
- Step 2: Infeed at feed rate F..
- Step 3: Feed stop at depth, spindle still rotating.
- Display of the message: «FACING COMPLETED? (Y):»
- Step 4: Operator action for facing.
- The operator answers the message by pressing Y (yes) to enable continuation to the next step (5).
- Step 5: Retraction at rapid in the tool axis.

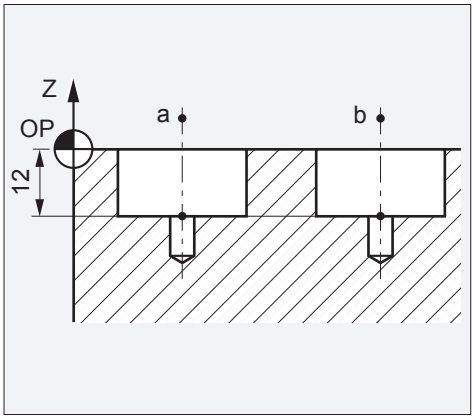
The message «FACING COMPLETED? (Y):» included in the cycle deletes the message, if any, programmed by address «$0».

Example

```
N.. ...
N50 G00 Xa Ya Za
N60 G88 Z-12 F30
N70 Xb Yb
N80 G80 G00 Z..
N.
```

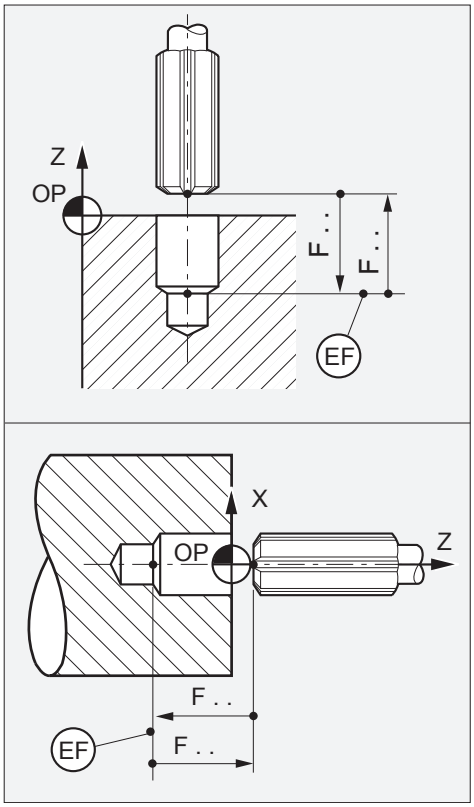
or

```
N.. ...
N60 G88 Xa Ya ERa Z-12 F30
N70 Xb Yb
N80 G80 G00 Z..
N..
```



5.12.13 Boring Cycle with Dwell at the Bottom of the Hole ► G89

G89	Boring cycle with dwell at the bottom of the hole.



Syntax (XY plane)

N.. [G17] **G89** [X.. Y..] Z.. [ER..] [EH..] [EF..] [F..]

G17	XY plane selection.
G89	Boring cycle with dwell in the bottom of the hole.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
EF..	Dwell expressed in seconds (maximum 99.99 s, format EF022, default EF = 1 second).
F..	Feed rate in the cycle.

Property of the Function

Function G89 is modal.

Cancellation

Function G89 is cancelled by one of functions G31, G80-G88.

Cycle Steps

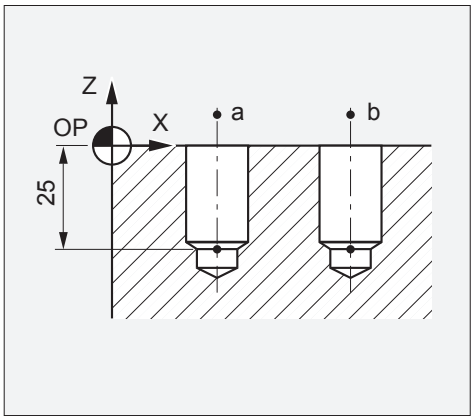
- Step 1: Rapid positioning in the plane.
- Step 2: Infeed at feed rate F..
- Step 3: Dwell at the end of boring.
- Step 4: Rapid retraction in the tool axis.

Example

```
N.. ...
N50 G00 Xa Ya Za
N60 G89 Z-25 EF1 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G89 Xa Ya ERa Z-25 EF1 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```



Syntax

N.. **G89** X.. / Z.. [ER..] [EH..] EF..	
G89	Boring cycle with dwell at the bottom of the hole.
X.. / Z..	End point on the machining axis.
ER..	Dimension of the retraction plane on the machining axis.
EH..	Impact plane in the machining axis.
EF..	Mandatory dwell expressed in seconds (maximum 99.99 s, format EF022, the default value of EF = 1 second).

Property of the Function

Function G89 is modal.

Cancellation

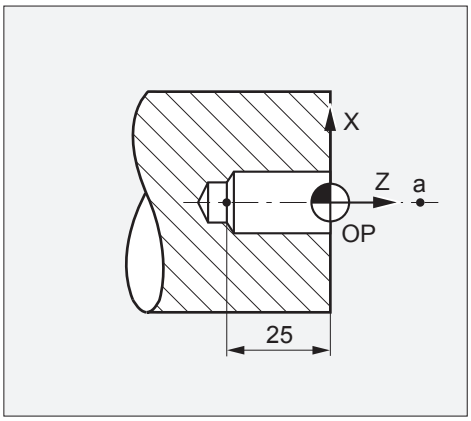
Function G89 is cancelled by one of functions G80-G85, G87 or functions G64, G65, G66.

Cycle Steps

- Step 1: Fast positioning in the machining axis.
- Step 2: Infeed at the feed rate F..
- Step 3: Dwell at the end of boring
- Step 4: Outfeed at the feed rate F.. along the tool axis.

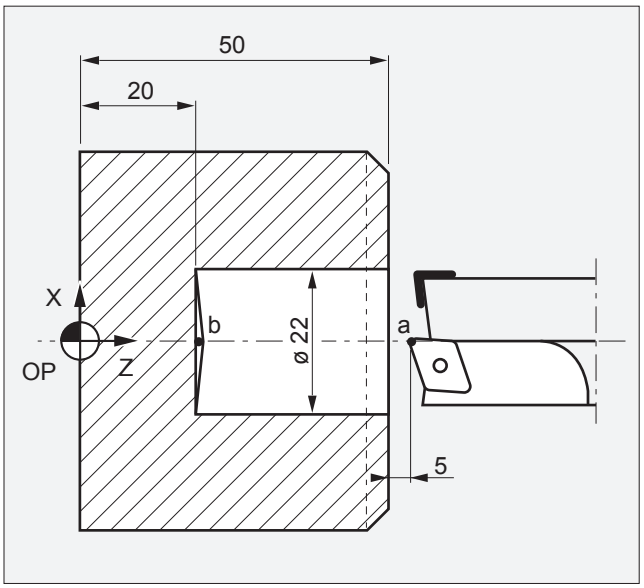
Example

```
N.. ...
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G89 Z-25 EF1
N80 G80 G00 X.. Z..
N..

or

N..
N50 G94 (or G95) F..
N60 G89 Xa ERa Z-25 EF1
N70 G80 G00 X.. Z..
N..
```



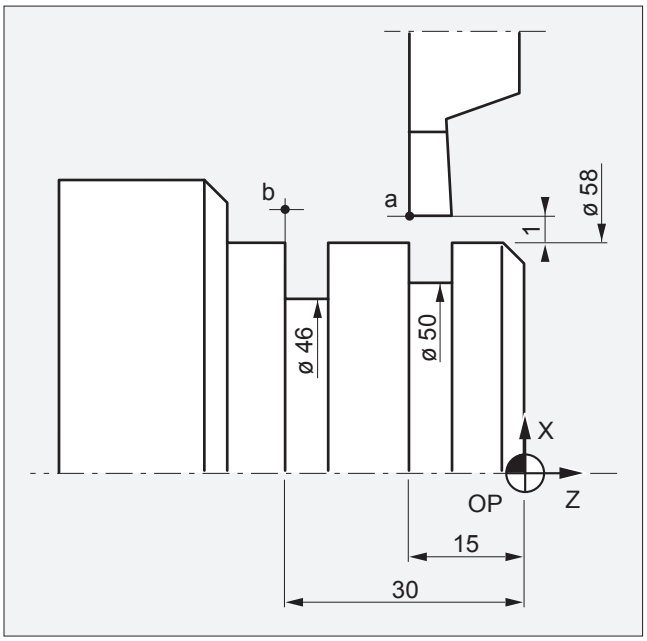
Examples of Programming Cycles

Drilling in the Z axis (cycle G83 or G87)




```
%62
N10...
N.. G92 S3500
N.. ...
N120 T06 D06 M06 (CARBIDE DRILL D=22)
N130 G97 S900 M40 M03
N140 G00 X0 Z55                          Point a
N150 G96 X22 S100
N160 G95 F0.15
N170 G83 (or G87) Z20 P10 Q5             Point b, cycle
N180 G97 S900
N190 G80 G00 G52 X.. Z..                 Cycle cancellation
N..
```

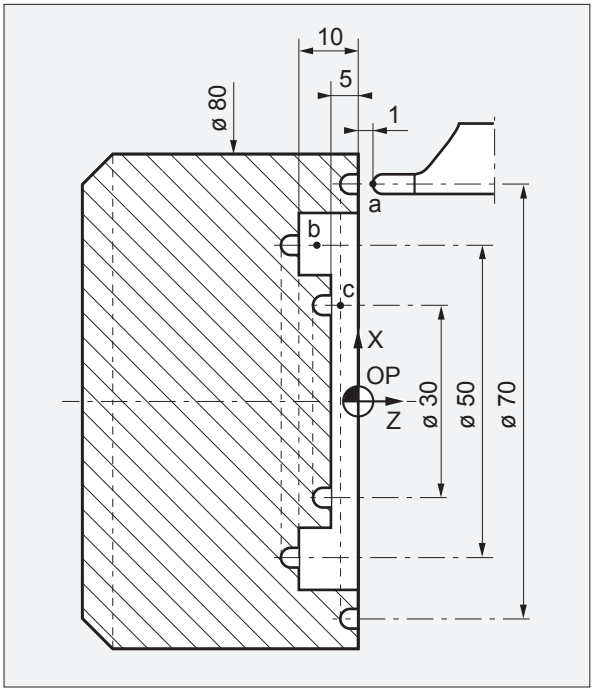
Execution of two grooves in the X axis (cycle G83 or G87)




```
%56
N10...
N.. G92 S3000
N..
N.. ...
N220 T08 D08 M06 (GROOVING TOOL L=4)
N230 G97 S900 M40 M04
N240 G16 P+                          Tool axis orientation on X+
N250 G00 X60 Z-15                    Point a
N260 G96 S50
N270 G95 F0.1
N280 G83 (or G87) X50 P2 Q1 G04 F1   Cycle
N290 X46 Z-30                        Point b, cycle repeat
N300 G97 S900
N310 G80 G16 R+ G00 G52 X.. Z..      Cycle cancellation
N..
```

Execution of three front grooves along the Z axis (cycle G82) with positioning on different levels using ER

Grooves with radius 3, depth 3.

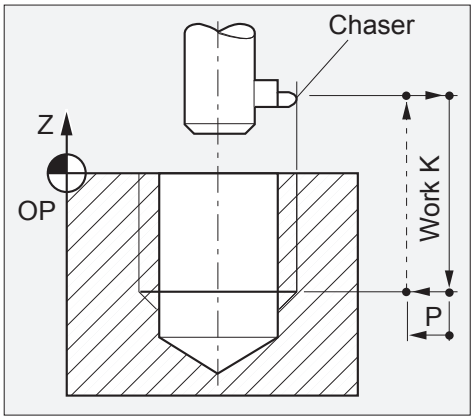



```
%72
N10...
N.. G92 S200
N..
N..
N190 T06 D06 M06 (GROOVE CUTTER R3)
N200 G97 S900 M04 M40
N210 G00 Z10
N220 G96 X80 S40
N230 G95 F0.08
N240 G82 X70 Z-3 ER1 EF2                Point a, cycle
N250 X50 Z-13 ER-9                      Point b, cycle
N260 ER-4
N270 X30 Z-8 EF1                        Point c, cycle
N280 G97 S900
N290 G80 G00 G52 X.. Z..                Cycle cancellation
N..
```

5.12.14 Thread Chasing Cycle ► G31

G31	Thread chasing cycle.

This cycle couples the tool feed rate to the spindle rotation.



Syntax (XY plane)

N.. [G17] [M03/M04] [S..] **G31** [X.. Y..] Z.. [ER..] [EH..] K.. P.. [F..] [EF..] [EC..]

G17	XY plane selection.
M03/M04	Spindle rotation.
S..	Spindle rotation speed.
G31	Thread chasing cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Approach or retraction dimension on the machining axis.
EH..	Impact plane in the machining axis.
K..	Thread pitch in mm. K = pitch with XY plane (G17) J = pitch with ZX plane (G18) I = pitch with YZ plane (G19).
P..	Absolute tool retraction dimension at the end of thread cutting.
F..	Number of thread starts (1 to 9, default 1).
EF..	Dwell time in seconds (maximum 99.99 s, format EF022, default value equal to two spindle rotations).
EC..	Value of the indexing position (the default value of EC is the last indexing position programmed).

Property of the Function

Function G31 is modal.

Cancellation

Function G31 is cancelled by one of functions G80-G89.

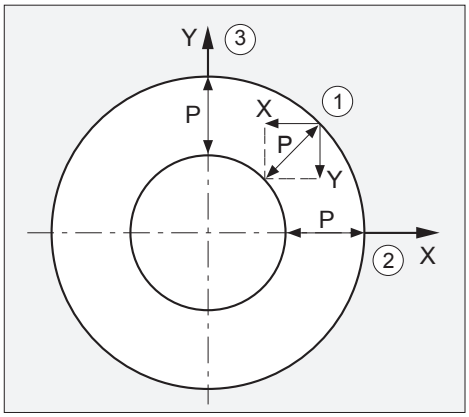
Notes

G31 needs encoder feedback from the spindle. The definition of this device (number of lines per revolution) is established by the machine manufacturer.

The crossing of an angular position (monitored by the system) starts movements for thread cutting. With block continuation, it is possible to change the plane and machining axis.

Clearance at the end of thread cutting

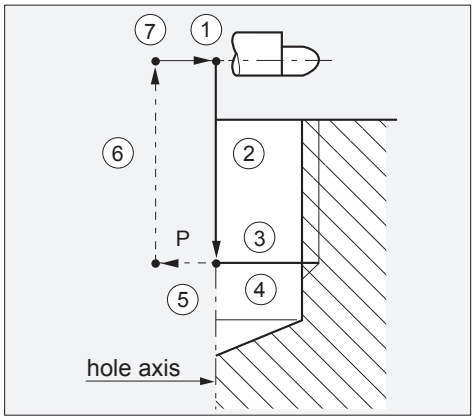
- Position 1 in the figure:
  Clearance along the two axes at 45 degrees,
  in this case X = Y = $\frac{P}{\sqrt{2}}$
- Position 2 on the figure:
  Clearance along the X axis. In this case, X = P.
- Position 3 in the figure:
  Clearance along the Y axis. In this case, Y = P.



Movements P are the same in the three planes (XY, ZX, YZ).

Cycle Steps

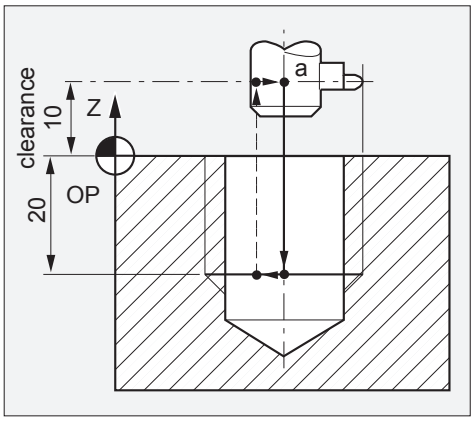
- Step 1: Rapid positioning of the chaser-holder to the centre of the hole then dwell for 1.2 seconds to start spindle rotation.
- Step 2: Infeed at the pitch programmed.
- Step 3: Dwell or two spindle rotations at the end of thread cutting.
- Step 4: Indexed spindle stop at position zero of the position encoder.
- Step 5: Rapid clearance by the value P in the axis of the plane and direction defined by the machine manufacturer.
- Step 6: Rapid retraction in the tool axis.
- Step 7: Rapid repositioning of the tool to the hole centre and restart of spindle rotation.



For thread cutting with several starts, the cycle is repeated after changing the angular position for each start.

Example

```
N..
N60 T09 D09 M06 (CHASER R2)
N50 G00 X.. Y.. Za
N60 G31 Xa Ya Z-20 K3 F2 P4 EF1
N70 G80 G00 Z..
N..
```



5.12.14.1 Cycle repeat to achieve the thread precision

```
N..
N.. G00 X.. Y.. Z..                          Chaser-holder approach
N100 S300 M40 M03 G31 X.. Y.. Z.. K.. P..    Cycle
N110 G00 Z200 M00 $0 CONTROLE                Retraction and spindle stop and display of a
                                             message (see chapter 5.20)
/N120 G79 N100 G04 F0.5
N130 ...
N..
```

Retraction and spindle stop in N110

Check of thread cutting precision.

If thread cutting is correct:

- Confirm by pressing the block skip key «/» on the NC and restart the cycle.
- The system skips block N120 and goes to block N130 (do no forget to inhibit block skip «/» for the rest of the program).

If thread cutting is not correct:

- Adjust the chaser. Do not press the block skip key «/».
- Restart the cycle. When reaching block N110, the system jumps to block N100 and repeats the cycle.
- At the end of block N100, select one of the two procedures after checking thread cutting precision again.

It is necessary to program a dwell in N120 so that the cycle can be repeated.

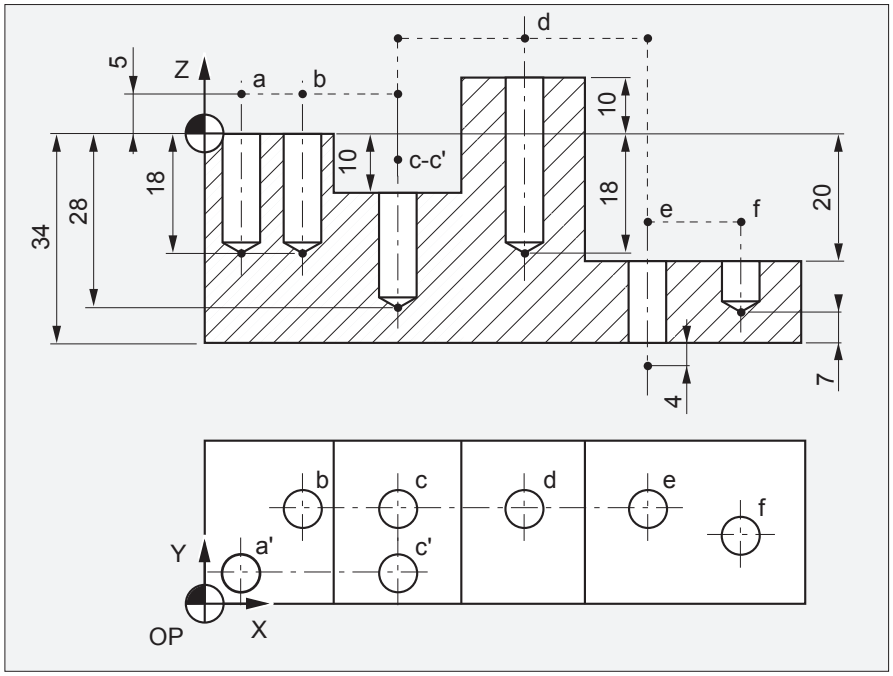
5.12.15 Cycle summary

CYCLES / BREAKDOWN OF MOVEMENTS	G31	G81	G82	G83	G84	G85	G86	G87	G88	G89
	Thread chasing	**Centre drilling**	**Counter boring**	**Peck drilling**	**Tapping**	**Reaming**	**Boring with indexed spindle stop**	**Drilling with chip breaking**	**Boring and facing**	**Boring with dwell in bottom of hole**
Downward	Work-infeed synchronised to the spindle	Work	Work	Rapid then work with n consecutive penetrations (P Q)	Work	Work	Work	Work with in consecutive penetrations (P Q)	Work	Work
Number of constant infeeds				Programmed by EP				Programmed by EP		
Retraction after infeed				Rapid						
Clearance after pecking				Programmed by EP						
Backoff between 2 infeeds								Programmed by EP		
Dwell on each infeed				Programmed by EP				Programmed by EP		
HOLE BOTTOM		Dwell	Programmed by EF	Programmed by EF on combined machine used for turning	Programmed by EF					Programmed by EF
		Spindle			Rotation reversal		Indexed stop programmed by EC			
		Retraction in the plane					Fixed or programmed by EP			
Upward Rapid		Rapid	Rapid	Rapid	Work	Work rate or other feed rate if EF present	Rapid	Rapid at the end of penetration	Rapid after confirmation by the operator	Work
End of upward travel					Spindle rotation reversal		Rapid return of the X and/ or Y axes in the boring axis and reset of the spindle to the initial state			

5.12.16 Examples of G81 to G89 cycles

Sequence of drilled holes in the XY plane

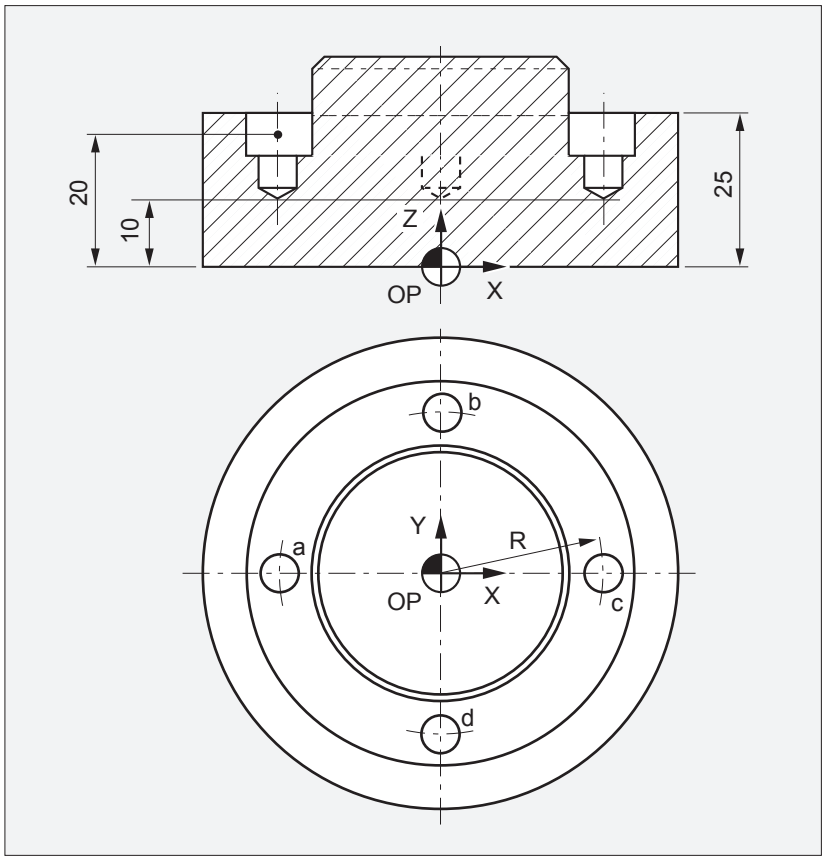
Drilling to different depths and positioning of the tool on different levels with ER.




```
%56
N10 G00 G52 Z..
N20 T12 D12 M06 (DRILL)
N30 S1000 M40 M03
N40 G00 Xa Ya Z50                    Tool approach
N50 G81 Z-18 ER5 F80                 Drilling cycle
N60 Xb Yb                            Cycle
N70 Xc ER-7 Z-28                     Cycle
N80 Yc'                              Cycle
N90 ER12
N100 G83 Xd Yd Z-18 P7 F100          Peck drilling cycle
N110 G81 Xe ER-17 Z-38               Drilling cycle
N120 Xf Yf Z-28                      Cycle
N130 G80 G00 G52 Z..
N..
```

Sequence of drilled holes in the XY plane

Drilling four holes with positioning by circular interpolation inside a groove.

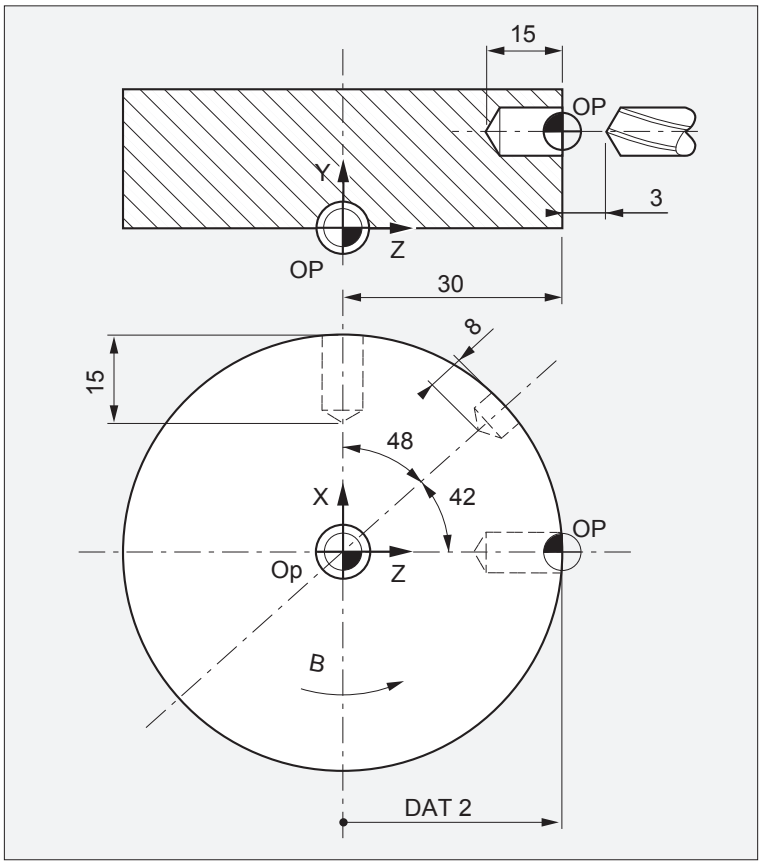



```
%45
N10 G00 G52 Z..
N20 T02 D02 M06 (DRILL)
N30 S1000 M40 M03
N40 G81 Xa Ya ER20 Z10 F80          Drilling cycle
N50 G02 Xb Yb I0 J0 (or R)          Cycle
N60 Xc Yc I0 J0 (or R)              Cycle   (Note the repetition of I0 and J0)
N70 Xd Yd I0 J0 (or R)              Cycle
N80 G00 G80 G52 Z..
N..
```

Sequence of drilled holes in the XY plane on a machining center with rotary axis B

Origin offset on Z.

Drilling to different depths, positioning by rotating the B axis.




```
%89
N10 G00 G52 Z..
N20 T08 D08 M06 (DRILL)
N30 S900 M40 M03
N40 G00 X0 Y0 Z3 B0
N50 G82 Z-15 EF1.5 F70          Drilling cycle
N60 B42 Z-8                     Cycle
N70 B90 Z-15                    Cycle
N80 G80 G00 G52 Z..
N..
```

5.13 Turning specific canned cycles

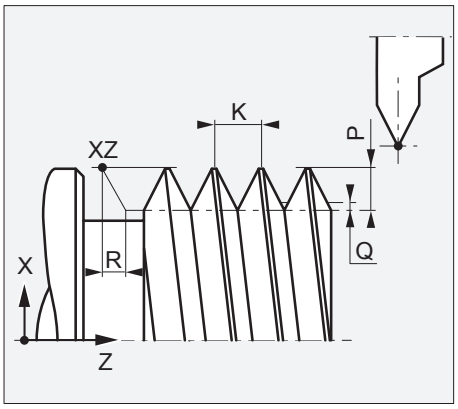
5.13.1 Constant Pitch Thread Chasing Cycle ► G33

G33	Constant pitch thread chasing cycle.

This function is used for cutting parallel, tapered or scroll threads.

The threads can be single or multiple start and can be cut by straight or angular penetration.

The consecutive passes are carried out with reducing depths of cut.



Syntax

N.. **G33** X.. Z.. K..[EA..][EB..]P..[Q..][R..][F..][S..] / [ES..]

G33	Constant pitch thread chasing cycle.
X.. Z..	Tool position at the end of thread cutting with respect to the program origin.
K..	Pitch in the thread cutting axis (X or Z) in mm.
EA..	Angle for tapered thread (see Fig. 5). - The default value of EA = 0: parallel thread cutting - EA90: scroll thread cutting.
EB..	Tool penetration angle (see Figs. 3 and 4) - The default value of EB = 0: straight penetration. - EB > 0: penetration down the thread flank in the direction of machining. - EB < 0: penetration down the thread flank opposite the machining direction.
P..	Total thread depth (including Q).
Q..	Depth of the final sizing pass (included in P) - Default no sizing pass (omit Q..). - Q = 0: spring cut required.
R..	Length of the tool retraction slope at the end of a pass (in the thread cutting axis) (see Figs. 1 and 2). - The default value of R = 0: retraction perpendicular to the thread cutting axis.
F..	Number of thread starts (format F01) (see Fig. 6). - Default single start.
S../ES..	Number of passes (format S02 or ES02) (not including pass Q), default 1 pass: - S..: Passes with reduced depth of cut, - ES..: Constant value passes.

Property of the Function

Function G33 is nonmodal.

Cancellation

Function G33 is cancelled at the end of the block.

Notes

The modal functions present before the G33 cycle are restored after execution of the cycle.

The thread pitch is applied to the thread cutting axis (axis on which the movement is largest).

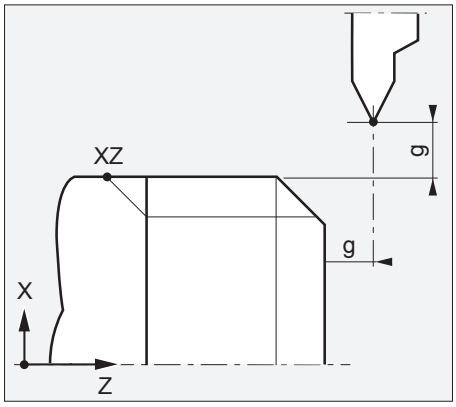
During the cycle, spindle speed override is inhibited (value forced to 100 per cent).

The maximum possible pitch is limited by the feed rate in mm/min acceptable by the machine (feed rate = pitch x rotation speed). Refer to the machine manufacturer's documentation.

Thread cutting start point

The tool must be backed off from the material by a clearance (g).

The clearance in the axis of penetration is mandatory. If the tool positioning and end of thread cutting values in the axis of penetration are the same, the system will generate an error message.



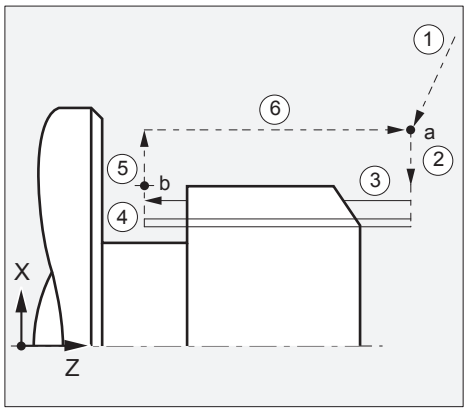
Cycle Steps

The steps below are given for information. The thread cutting cycle contains only the main parameters.

```
N.. ...
N220 G00 Xa Za                    Rapid to start of thread cutting
N230 G33 Xb Zb K.. P.. S..
N..
```

- Step 1: Tool positioning in X and Z to the thread cutting start point (point a).
- Step 2: Rapid advance in the axis of penetration (first pass).
- Step 3: Execution of the first pass in the thread cutting axis Z (point Zb).
- Step 4: Retraction along X (point Xb).
- Step 5: Rapid retraction of the penetration axis.
- Step 6: Rapid return to the thread cutting start point.

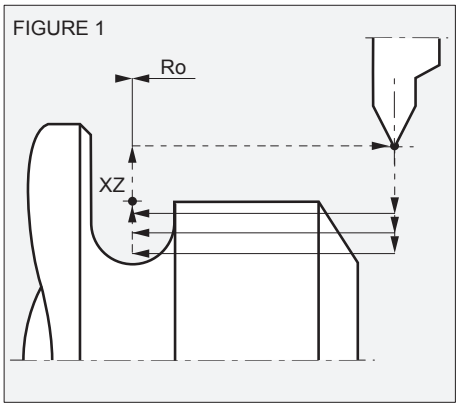
Successive passes are executed in the same way as steps 2 to 6 and may be followed by a spring pass.



5.13.1.1 Tool retraction slope at the end the pass defined by R

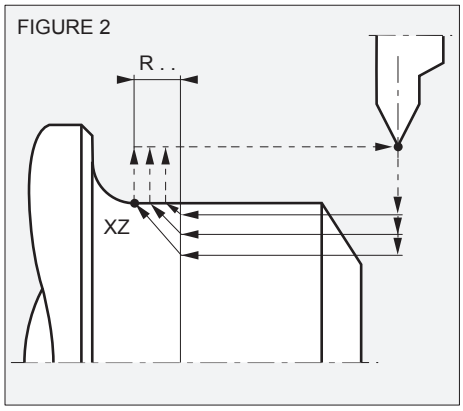
Thread cutting without retraction slope

FIGURE 1



Thread cutting with retraction slope

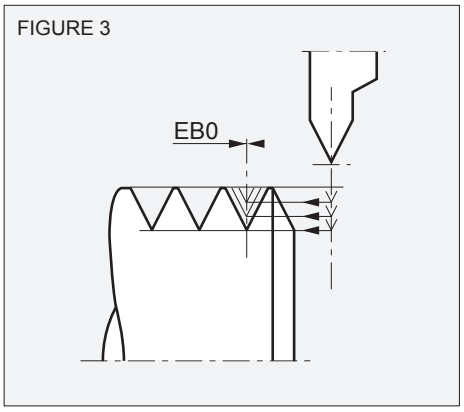
FIGURE 2



5.13.1.2 Penetration defined by EB

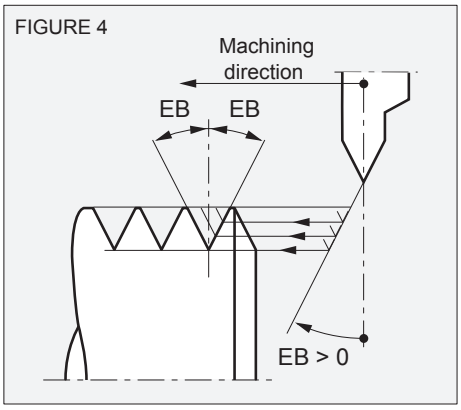
Thread cutting by straight penetration

FIGURE 3



Thread cutting by penetration down the thread flank

FIGURE 4



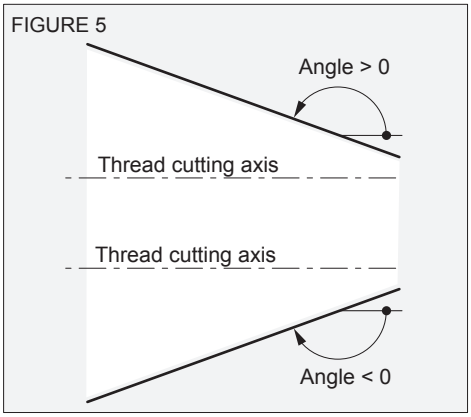
5.13.1.3 Tapered thread cutting, angle defined by EA

The value of EA is positive in the counterclockwise direction, modulo 180 degrees.

i.e. -45 deg < EA < 45 deg for:
- Z thread cutting axis
- X penetration axis.

i.e. EA > 45 deg or EA < -45 deg for:
- Z axis of penetration
- X thread cutting axis.

FIGURE 5

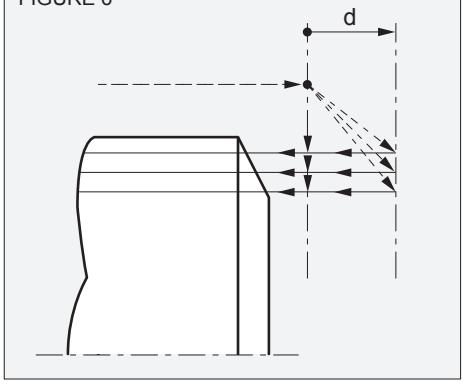


5.13.1.4 Multiple start thread cutting

The offset «d» (pitch) corresponding to each start is in the direction opposite the thread cutting direction.

Each start will be completed at one depth before continuing to the next pass depth.

FIGURE 6



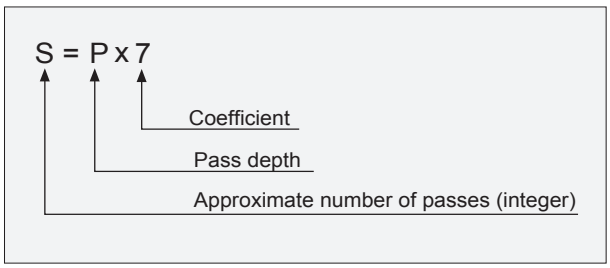
Reminders

Determination of the pass depth (P) for ISO threads.

External thread: 0.613 x pitch

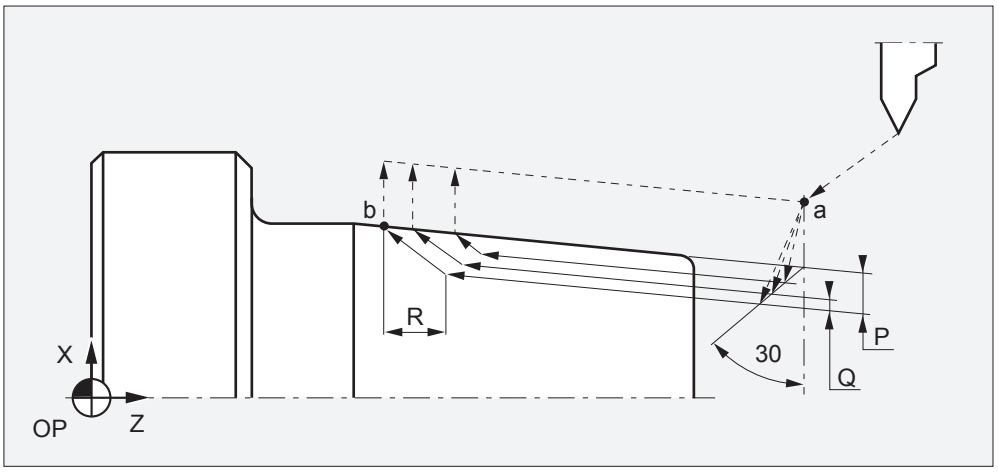
Internal thread: 0.577 x pitch

Determination of the approximate number of passes (S)



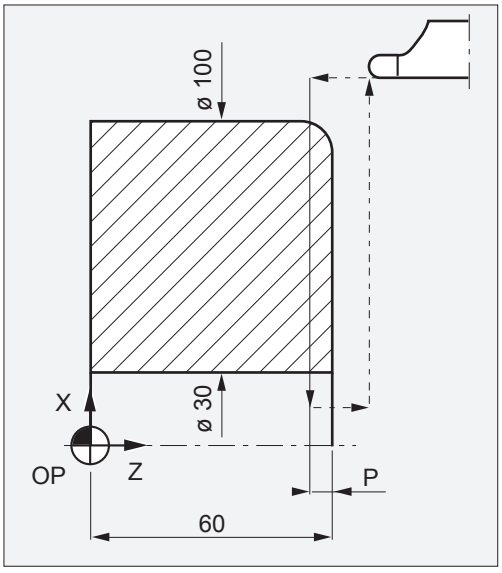
Examples

Cutting of external tapered thread (5 deg)




```
N.. ...
N140 T09 D09 M06 (THREAD CUTTING TOOL, PITCH=1 RH THREAD)
N150 G97 S1000 M40 M03
N160 G00 Xa Za                                    Start of thread cutting
N170 G33 Xb Zb K1 EA175 EB30 P0.61 Q0.02 R4 S5
N.. ...
```

Cutting of front threads



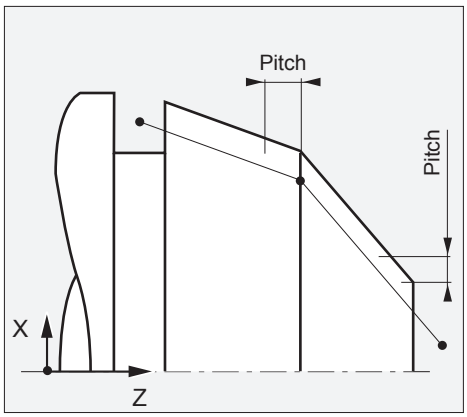

```
N.. ...
$ FRONT THREAD ALONG X
N140 T05 D05 M06 (TOOL R=2)
N150 G97 S200 M40 M03
N160 G00 X110 Z64                          Point a, start of thread cutting
N170 G33 X20 Z60 K4 EA90 P2 Q0 S14
N..
```

5.13.2 Sequenced Thread Cutting ► G38

G38	Sequenced thread cutting.

This functions allows the execution of several consecutive thread cutting blocks.

The threads can be cylindrical or tapered.



Syntax

N.. **G38** X.. Z.. K..

G38	Sequenced thread cutting.
X.. Z..	Tool position at the end of thread cutting with respect to the program origin.
K..	Pitch in the thread cutting axis (X or Z) in mm.

Property of the Function

Function G38 is modal.

Cancellation

Function G38 is cancelled by one of functions G00, G01, G02 or G03.

Notes

Different pitches (K) can be programmed in consecutive thread cutting blocks with G38.

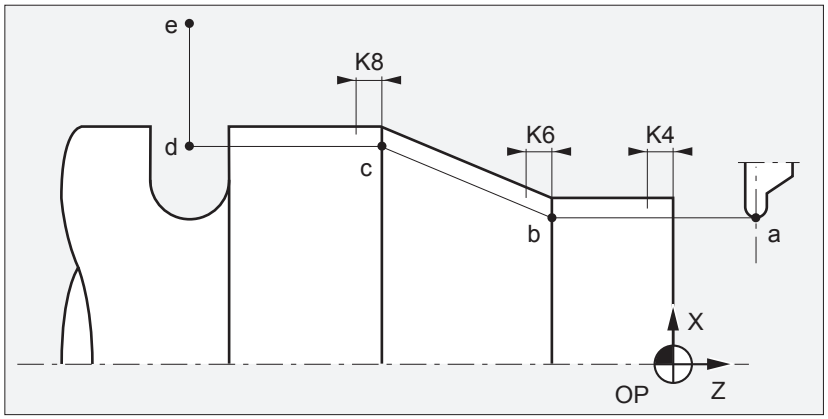
The thread pitch is applied to the thread cutting axis (axis on which the movement is largest).

The consecutive passes, to obtain the required depth, can be programmed by subroutine calls. (During execution of the cycle, spindle speed override by potentiometer is inhibited (value forced to 100 percent).

Function G38 must be programmed:
- without miscellaneous functions (M) in the cycle blocks,
- without use of constant surface speed (nor retract in step for successive passes).

The maximum possible pitch is limited by the feed rate in mm/min acceptable by the machine (feed rate = pitch x rotation speed). Refer to the machine manufacturer's documentation.

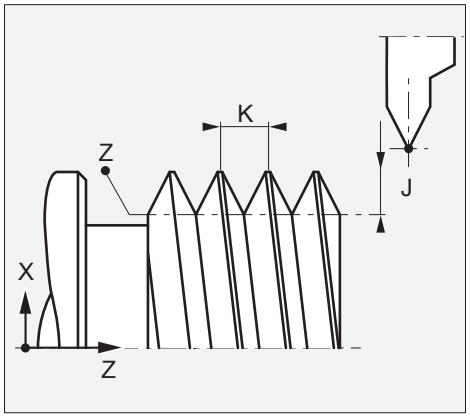
Example




```
N.. ...
N100 T04 D04 M06 (TOOL R=2)
N110 G97 S400 M40 M03
N120 G00 Xa Za                  Approach point
N130 G38 Xb Zb K4               Cycle with 4 mm pitch
N140 Xc Zc K6                   Cycle with 6 mm pitch
N150 Xd Zd K8                   Cycle with 8 mm pitch
N160 G00 Xe Ze                  Cycle cancelled
N..
```

5.13.3 Resume Thread Cutting ► G38+

G38+ Resume Thread Cutting .



Thread cutting may be interrupted and the part removed from the chuck, for instance to check it. The same part may then be reinstalled in the chuck to continue the thread cutting operation (possibly using a different spindle speed than during roughing).

When reinstalled in the chuck, the part is in a different angular position (and possibly a different axial position) than before.

The resume thread cutting function has been created to teach in the new position of the thread and use it as argument in a resume thread cutting cycle.

Teach-in of the thread position is carried out manually by positioning the tool in the roughed thread and measuring the axial position) (or radial position for thread cutting in X) called Xpos_teachin as well as the angular position of the spindle Spos_ teachin (E90100 for spindle 1).

Syntax

G38+ J<Xpos_delta> EC<Spos_teachin> K.. Z..

J<Xpos_delta> = <Xpos_teachin> - <Xstart>	Difference between the teach-in position of the axis and its start point when thread cutting is resumed, expressed in programming units (mm or inches). For a taper thread, the difference is that of the major axis, i.e., the axis to which the pitch applies.
EC<Spos_teachin> Teach-in	Angular spindle position in degrees.
K..	Thread pitch.
Z..	Thread cutting end point.

Notes

Arguments J and EC must be programmed immediately after function G38+.
When the resumed thread cutting cycle includes several passes with straight or oblique infeed, the values of arguments J and EC remain unchanged.

For sequenced thread cutting, the blocks after the initial block must not include any arguments except the end point and possibly the pitch.

Example

Example of longitudinal thread cutting with tapered retraction

`G0X..Z..`	Positioning at the start point
`G38+ J.. EC.. K.. Z..`	Longitudinal thread cutting
`X.. Z..`	Tapered sequencing of the thread
`G0 X..`	Rapid retraction off the thread.

Constraints

High-speed machining is prohibited for resumed thread cutting to allow on-the-fly exit from the thread. If this function is active when G38+ is analysed, it is automatically cancelled (E11012 = 0).

The gains on the axes must be set consistently with the position loop time constant set in machine parameter P56.

The spindle must be rotating at a steady speed before the block containing G38+ is executed.

5.13.3.1 Inclusion of function G38+ in Cycle G33

Syntax of Resumed Thread Cutting Included in the Call to Cycle G33

The teach-in positions are declared in EC.. for the spindle position and EZ.. or EX.. for the axis, depending on whether thread cutting is axial or radial.

The presence of these two arguments distinguishes between normal and resumed thread cutting.

G33 X.. Z.. P.. EZ.. EC.. K.. [S..] [R..] [EA..] [EB..] [F..]

The difference between the teach-in position of the axis and its initial position is calculated in macro %10033 as well as its assignment to function J after G38+.

Macro %10033 ensures that the syntax is consistent:

It checks that both the teach-in position of the spindle and the teach-in position on the thread axis (major axis) are programmed. For a 45-degree taper thread, either axis is accepted.

In case of an error, it displays the message No 838: $ Restart threading incoherent data.

5.13.3.2 Associating a Teach-in Cycle with Resume Thread Cutting Cycle G33

An operator may visually note the positions to be taught in and then enter them manually in a G33 block.

It is also possible to call a subroutine in a program before the G33 cycle to read variables indicating the spindle/axis positions to be assigned to EZ and EC.

Example

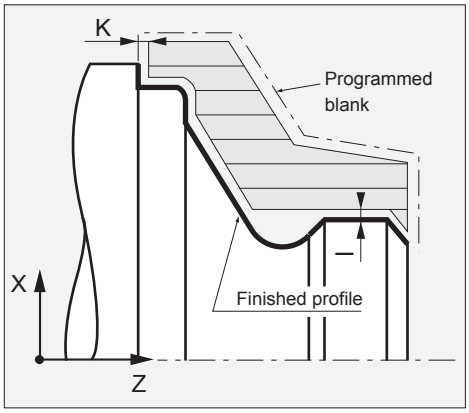
```
%1038                                          Teach-in resume thread cutting
VAR [b] = [.BM69]*3 + [.BM68] + [.BM68] + [.BM67]
[Bpom] = 91124+[b] [Bpos] = 90100+[b]
[ZTeach] [XTeach] [BTeach]
ENDV
M5
WHILE E[Bpom] = 1 DO G4 F.1 $                  Homing on spindle required
ENDW
M12 $                                          manually set position in thread
                                               After action on B.CYCLE, read position
[BTeach] = E[Bpos] /10000 [XTeach] = UE70000 [ZTeach] = UE72000
M12 $                                          Move off the thread
SAVE [BTeach]/[XTeach]/[ZTeach]                Store position
%10                                            Main longitudinal thread cutting program
...
G77 H1038                                      Call position teach-in
M4 S...
...
G0 Z... X...                                   Go to start of cycle
G33 Z... X.. K... P.. S… EZ[ZTeach] EC[BTeach]
...
```

5.13.4 Rough Turning/Facing Cycle ► G64

G64	Turning/Facing cycle.

This function is used to rough a volume of material located between the specified dimensions of a given blank profile and a finished profile.

This turning or facing cycle can be used for external or internal working.



Syntax

N.. **G64** [N.. N..] / [I.. K..] P.. / R..

N.. BLANK DEFINITION

G64	Turning/Facing cycle.
N.. N..	Numbers of the first and last block defining the finished profile (minimum two blocks, maximum 50 blocks).
I..	Finishing allowance in X.
	- The default value of I = 0.
K..	Finishing allowance in Z.
	- The default value of K = 0.
P.. / R..	Cut depth (unsigned).
	P: depth of cut in X (turning)
	R: width of cut in Z (facing)
N.. BLANK DEFINITION	Sequence of blocks defining the blank dimensions (these blocks are located between G64 and the cycle cancellation code G80).

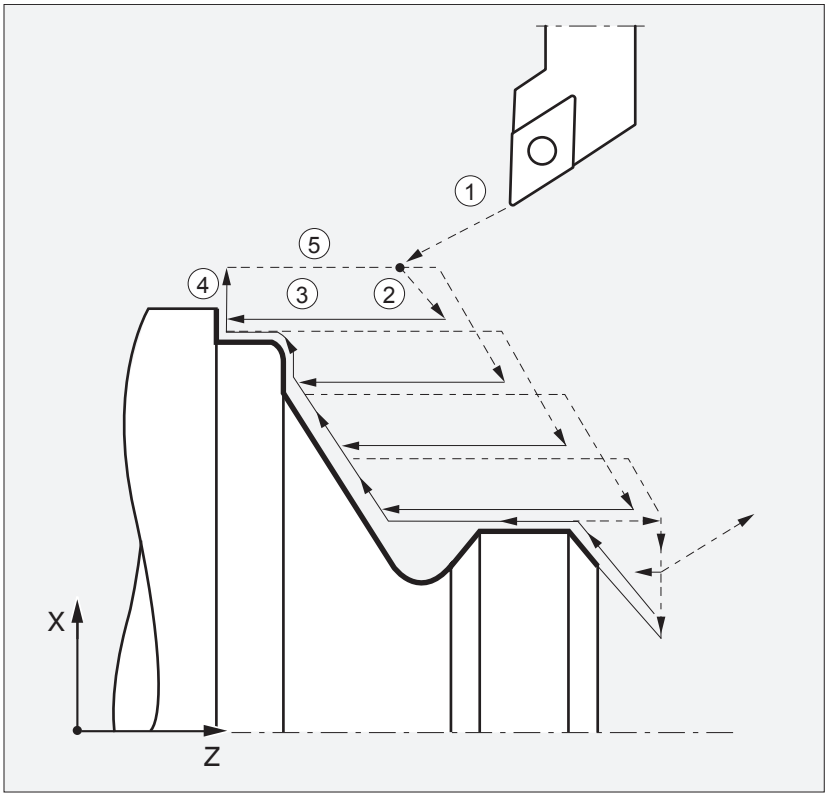
Property of the Function

Function G64 is modal.

Cancellation

Function G64 is cancelled by function G80.

Cycle Steps



- Step 1: Tool positioning in XZ.
- Step 2: Rapid to pass depth (to depth «P» for turning).
- Step 3: Execution of the first pass at the feed rate.
- Step 4: Lift along the profile at the feed rate.
- Step 5: Rapid return to the cycle start point.

Execution of the following passes in the same way as steps 2 to 5, then retraction.

Notes

When the cycle is programmed, the system must be in state G40 (G41 or G42 tool radius offset cancelled).

The feed rate and its argument can be programmed in the cycle block, e.g.:

```
N.. G64 N.. N.. I.. K.. P.. G95 F0.25
```

Notes on definition of the finished profile

Both first and last blocks N.. to N.. defining the finished profile dimensions must include both coordinates X and Z as they may be analysed in reverse order.

Addresses I and K must be signed according the direction of the machining allowance (e.g. «I-..» for a bore).

The cycle will not execute any grooves (in the face or on diameter) included in the definition of the finished profile.

At the end of the cycle, the system is left in G0 mode.

The finished profile definition can be placed before the cycle call, but in this case it is necessary to use branch function G79 (see chapter. 5.15).

The following program variables cannot be used in the finished profile definition. Otherwise, the system returns error message 96:
- L100-L199
- L900-L959 (see chapter 7.1).

(The only case where these variables can be used is when M999 is programmed, see chapter. 5.18.9)

Notes on the blank definition

The blank definition cannot contain:
- PGP (Profile geometry Programming) blocks,
- a block defining a curve.

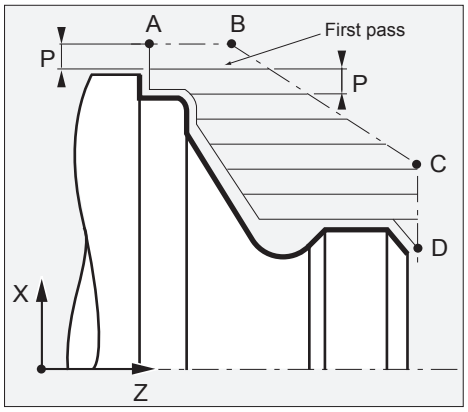
The blank definition blocks can include miscellaneous M functions.

5.13.4.1 Roughing cycle P or R

The roughing cycle uses P or R to determine turning or facing, plus the directional sense of increasing/decreasing dia. from the blank definition, to determine internal/external working.

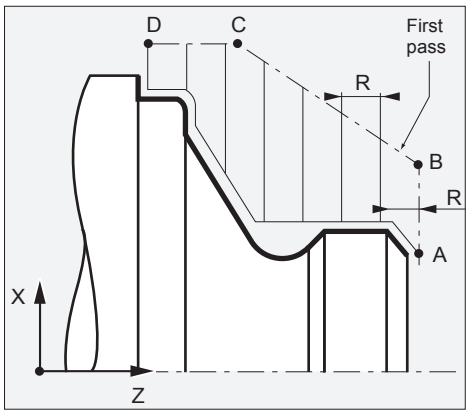
Rough turning

```
N200 ...
N210 G64 N.. N.. I.. K.. P4
N220 X.. Z..                  Point A
N230 Z..                      Point B
N240 X.. Z..                  Point C
N250 X..                      Point D
N260 G80 G52 X.. Z..
N..
```



Rough facing

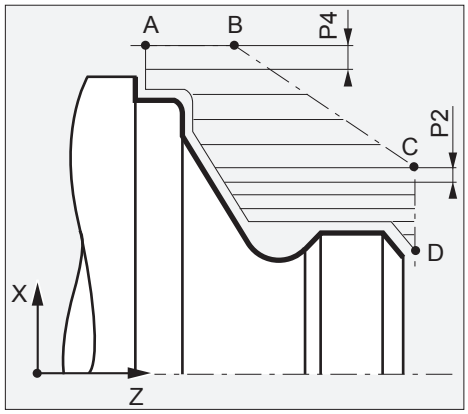
```
N200 ...
N210 G64 N.. N.. I.. K.. R4
N220 X.. Z..                  Point A
N230 X..                      Point B
N240 X.. Z..                  Point C
N250 Z..                      Point D
N260 G80 G52 X.. Z..
N..
```



The pass depth defined in the cycle block (P or R) can be modified during the blank definition blocks.

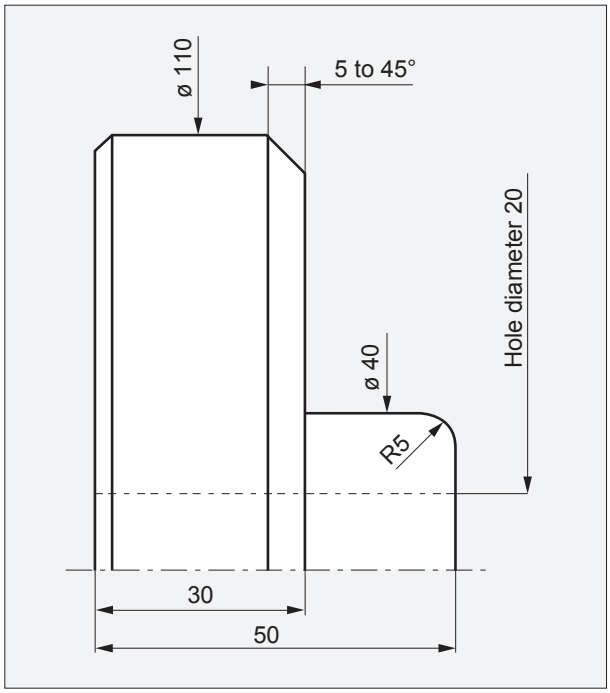
Example

```
N200 ...
N210 G64 N.. N.. I.. K.. P4
N220 X.. Z..                  Point A
N230 Z..                      Point B
N240 X.. Z..                  Point C
N250 X.. P2                   Point D,
                              new P
N260 G80 G52 X.. Z..
N..
```

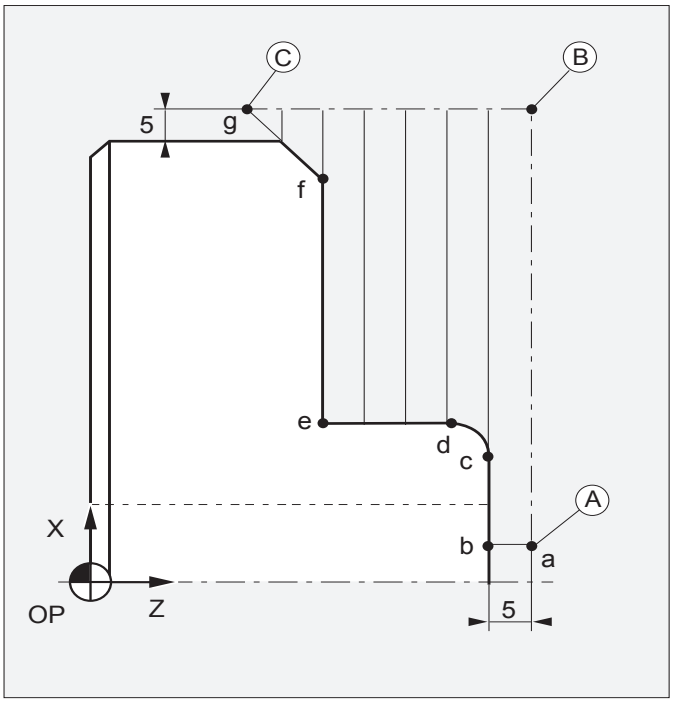


Examples

Roughing an external profile by facing



Machining paths




```
%46
N10 G00 G52 X.. Z..
N20 T01 D01 M06 (FACING TOOL R=0.8)
N30 S900 M40 M04
N40 X120 Z55
```

Tool change position

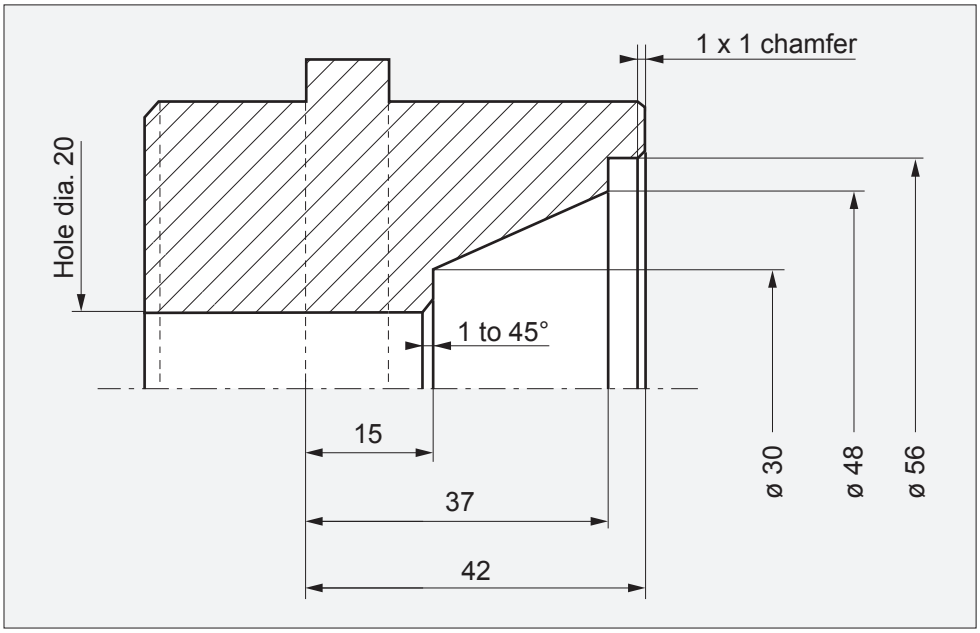
Point B, tool at the start of roughing

```
N50 G92 S3500
N60 G96 S200
N70 G95 F0.2
N80 G64 N140 N210 I0.4 K0.2 R3          Cycle
N90 X16 Z55                             Point (A)
N100 X120                               Point (B)   } Blank definition
N110 Z20                                Point (C)
N120 G80 G52 X.. Z.. G97 S900           Cycle cancellation
N130 T03 D03 M06 (COPYING TOOL R=0.4)
N140 G42 X16 Z55                        Point a, tool at the start of finishing
N150 G96 S250
N160 G01 Z50 F0.1                       Point b
N170 X30                                Point c
N180 G03 X40 Z45 R5                     Point d
N190 G01 Z30                            Point e
N200 X100                               Point f
N210 X120 Z20                           Point g
N220 G00 G40 G52 X.. Z.. G97 S900 M05
N230 M02
```

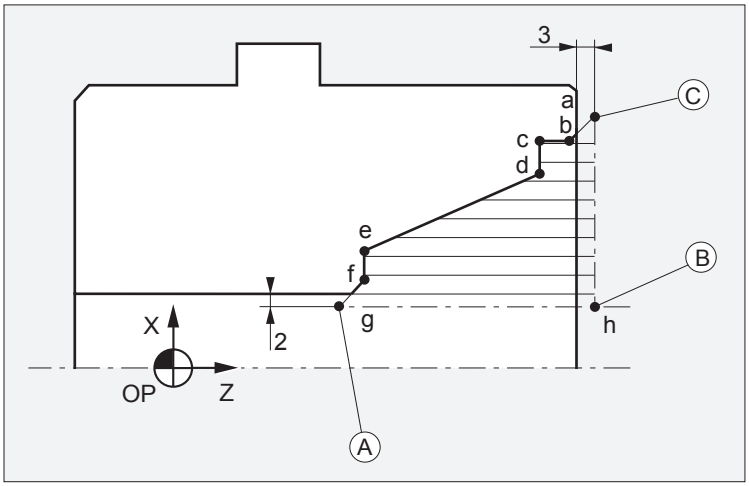
Changes required to program %46 for turning.

```
N80 G64 N140 N210 I0.4 K0.2 P3          Address R replaced by P

N90 X120 Z20
N100 Z55                                Blank definition reversed
N110 X16
```

Roughing of an internal profile by turning



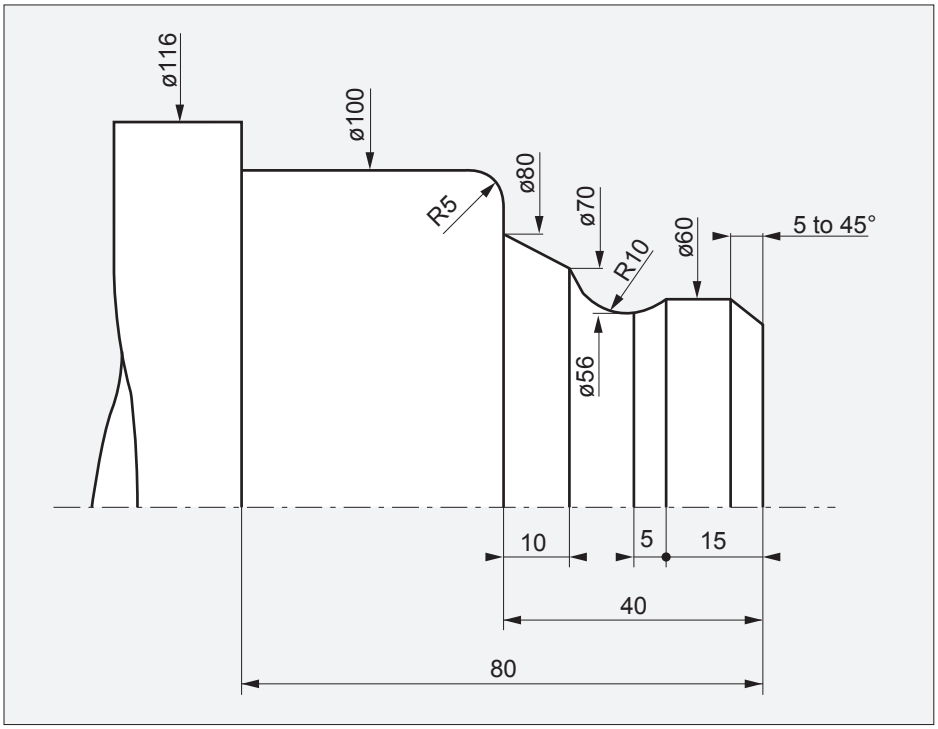
Machining paths



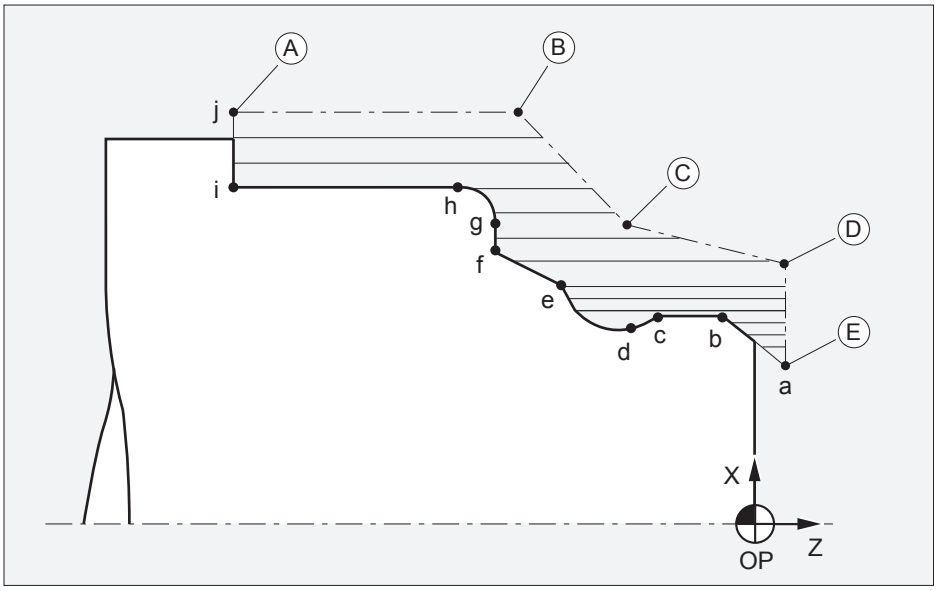

```
%48
N10 G00 G52 X.. Z..                          Tool change position
N20 T07 D07 M06 (BORING TOOL R=0.4)
N30 S900 M40 M04
N40 X18 Z45                                  Point B, tool at the start of roughing
N50 G92 S3500
N60 G96 S100
N70 G95 F0.15
N80 G64 N140 N210 I-0.2 K0.1 P2              Cycle
N90 X16 Z12                                  Point (A)
N100 Z45                                     Point (B)  } Blank definition
N110 X64                                     Point (C)
N120 G80 G52 X.. Z.. G97 S900                Cycle cancellation
N130 T09 D09 M06 (BORING TOOL R=0.2)
N140 G41 X64 Z45                             Point a, tool at the start of finishing
```

```
N150 G96 S120
N160 G01 X56 Z41 F0.08          Point b
N170 Z37                        Point c
N180 X48                        Point d
N190 X30 Z15                    Point e
N200 X22                        Point f
N210 X16 Z12                    Point g
N220 G00 Z45 G97 S900           Point h
N230 G40 G52 X.. Z.. M05
N240 M02
```

Roughing of an external profile by turning (preformed blank profile)



Machining paths




```
%62
N10 G00 G52 X.. Z..
N20 T01 D01 M06 (BIDIRECTIONAL TOOL R=0.8)
N30 S900 M40 M04
N40 X120 Z-30
N50 G92 S3500
N60 G96 S180
N70 G95 F0.25
```

Tool change position

Point B, tool at the start of roughing

```
N80 G64 N170 N270 I0.3 K0.2 P4
N90 X120 Z-80
N100 Z-40
N110 X80 Z-20
N120 X70 Z5
N130 X40 P2
N140 G97 S900
N150 G80 G52 X.. Z..
N160 T03 D03 M06 (COPYING TOOL R=0.4)
N170 G42 X40 Z5
N180 G96 S250
N190 G01 X60 Z-5 F0.1
N200 Z-15
N210 X56 Z-20
N220 G02 X70 Z-30 R10
N230 G01 X80 Z-40
N240 X90
N250 G03 X100 Z-45 R5
N260 G01 Z-80
N270 X120
N280 G00 G40 G52 X.. Z.. G97 S900 M05
N230 M02
```

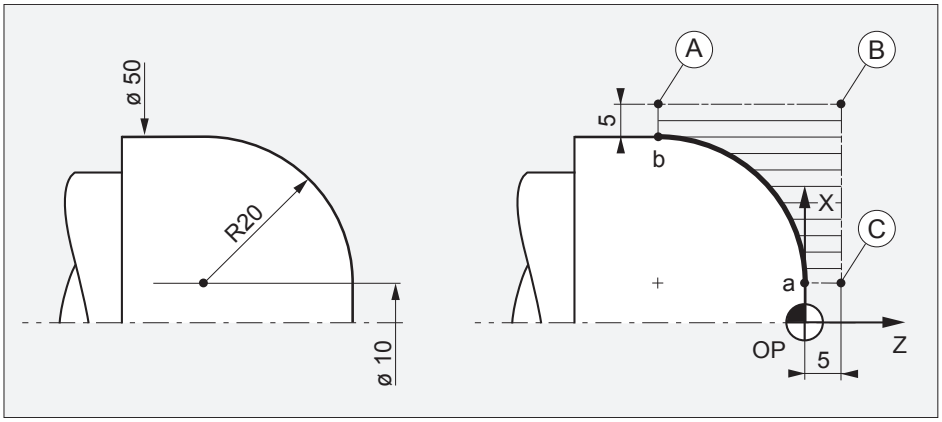
Cycle
Point (A)
Point (B)
Point (C)
Point (D)
Point (E)
} Blank definition

Cycle cancellation

Point a, tool at the start of finishing

Point b
Point c
Point d
Point e
Point f
Point g
Point H
Point i
Point j

Roughing then finishing by turning of a profile defined by two points (a and b)



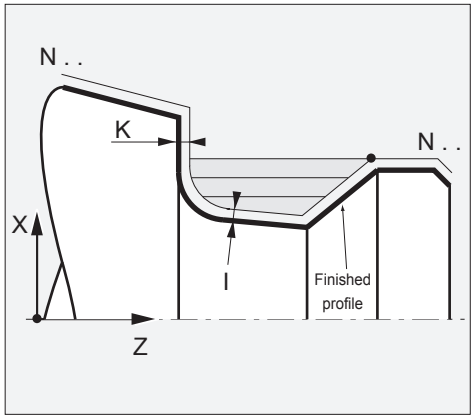

```
%64
N10 G00 G52 X.. Z..                      Tool change position
N20 T01 D01 M06 (TOOL R=0.8)
N30 S900 M40 M04
N40 X60 Z5                               Point B, tool at the start of roughing
N50 G96 S300
N60 G92 S3000
N70 G95 F0.3
N80 G64 N150 N160 P3 I0.2 K0.2           Cycle
N90 X60 Z-20                             Point A
N100 Z5                                  Point B  } Blank definition
N110 X10                                 Point C
N120 G80 G52 X.. Z.. G97 S900            Cycle cancellation
N130 T03 D03 M06 (OUTIL R=0.4)
N140 X10 Z2                              Approach point
N150 G42 G01 X10 Z0                      Point a
N160 G03 X50 Z-20 I10 K-20               Point b
N170 G01 G40 X54 G97 S900                Retraction point
N180 G00 G52 X.. Z.. M05
N190 M02
```

5.13.5 Groove Roughing Cycle ► G65

G65	Groove roughing cycle.

This function is used to rough a groove whose profile is programmed during definition of a finished profile.

The cycle executes axial grooves by turning and front grooves by facing.



Syntax

N.. **G65** [N.. N..] / X.. / Z.. [I.. K..] [EA..] P.. / R.. [Q..] [EF..]

G65	Groove roughing cycle.
N.. N..	First and last finished profile block numbers. The groove profile must be located between these blocks (maximum 50 blocks).
X.. / Z..	End of cut position in the appropriate axis. - X for front rough facing, - Z for axial rough turning.
I..	Finishing allowance in X. - The default value of I = 0.
K..	Finishing allowance in Z. - The default value of K = 0.
EA..	Cut introduction angle in the groove (see Fig. 1).
P.. / R..	Cut depth (unsigned). P: depth of cut in X (turning). R: width of cut in Z (facing).
Q..	Positioning clearance (see Fig. 2). - The default value of Q = 0 (approach distance without clearance at the machining feed rate).
EF..	Material penetration feed rate. - The default value of EF is the same as the modal feed rate F.

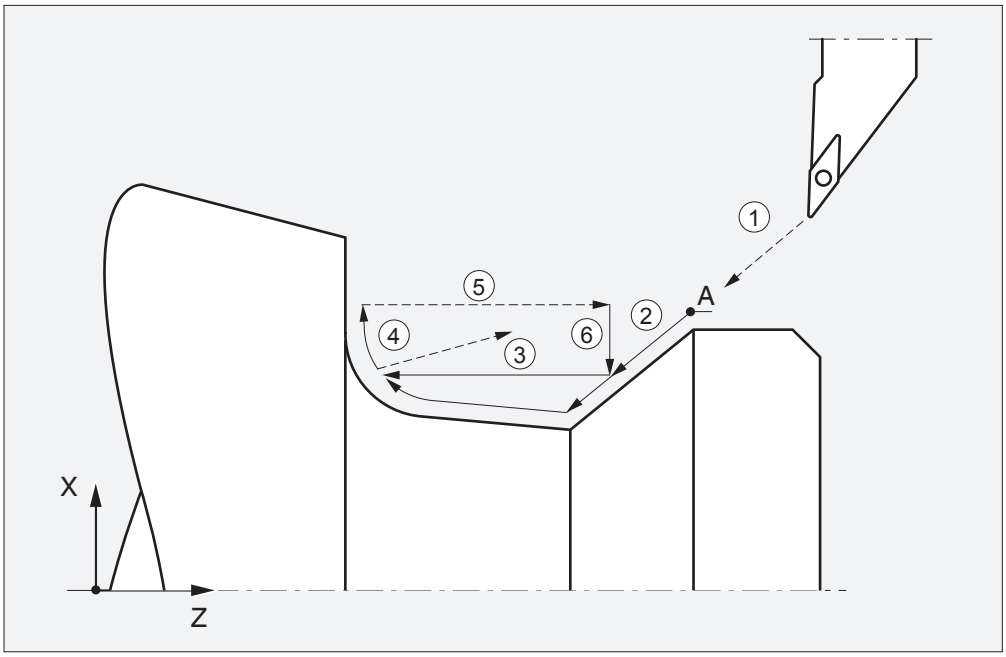
Property of the Function

Function G65 is nonmodal.

Cancellation

Function G65 is cancelled at the end of the block.

Cycle Steps



- Step 1: Tool positioning (point A, cycle start).
- Before the cycle start, the system takes into account the declared tool radius.
- Step 2: Infeed at the introduction angle at the machining feed rate (to depth P for turning).
- Step 3: Execution of the pass at the machining feed rate.
- Step 4: Retraction at the machining feed rate along the groove profile.
- Step 5: Rapid return to the perpendicular over the start point for the next pass.
- Step 6: Positioning at the machining feed rate (if Q with a value not programmed).

The following passes are executed in the same way as steps 2 to 6.

After execution of the last pass, the tool retracts at rapid to the cycle start point.

Notes on argument

Both the finished profile definition blocks N.. to N.. must include both the coordinates X and Z.

Tool positioning to the cycle start must be programmed in G40 (G41 or G42 radius offset cancelled).

The feed rate function and argument can be programmed in the cycle block, e.g.:

```
N.. G65 N.. N.. I.. K.. P.. G95 F0.1.
```

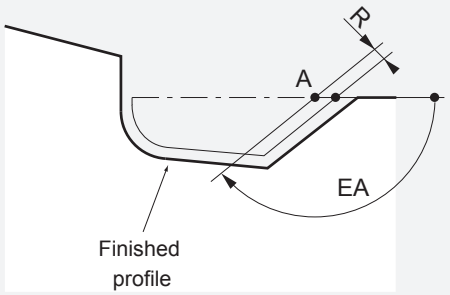
At the end of the cycle, the system is left in G0 mode.

Notes on argument EA

- EA: cut introduction angle defines a line from the cycle start point (point A) to the intersection with the finished profile.

After positioning to the start point, the system reads the declared tool radius (R) before executing the first pass.

FIGURE 1

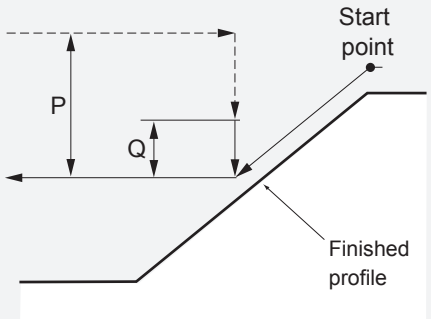


Notes on argument Q

After the first pass, when argument Q is programmed with its value, the approach distance before each new cut is carried out in two steps:

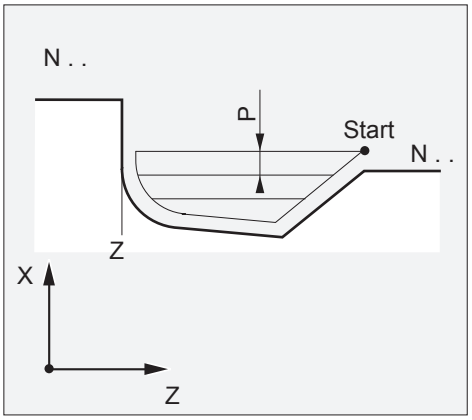
- At high speed up to value Q,
- At the machining feed rate until the start of the next pass.

FIGURE 2



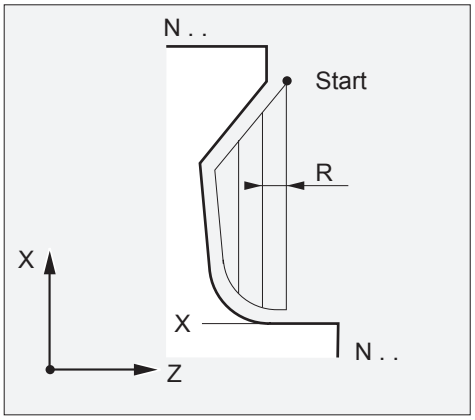
Diameter roughing

```
N70 ...
N80 G65 N.. N.. Z.. I.. K.. EA.. P..
N90 G52 X.. Z..
N..
```



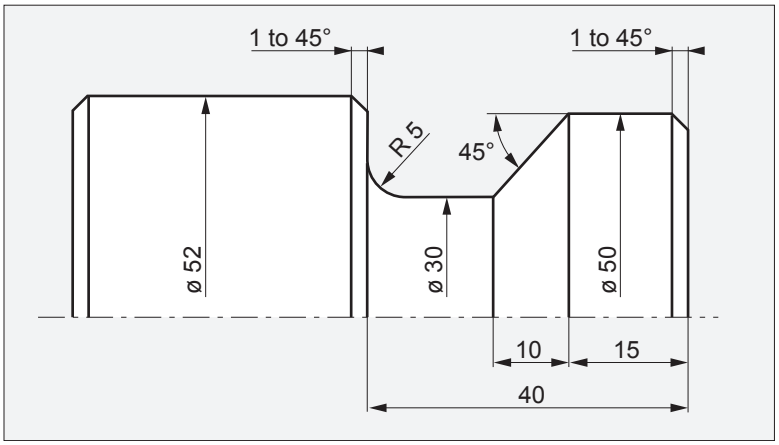
Face roughing

```
N70 ...
N80 G65 N.. N.. X.. I.. K.. EA.. R..
N90 G52 X.. Z..
N..
```

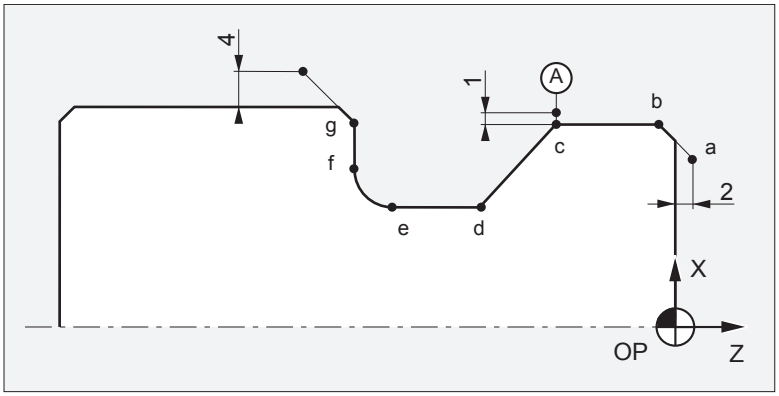


Examples

Roughing an axial groove



Machining paths

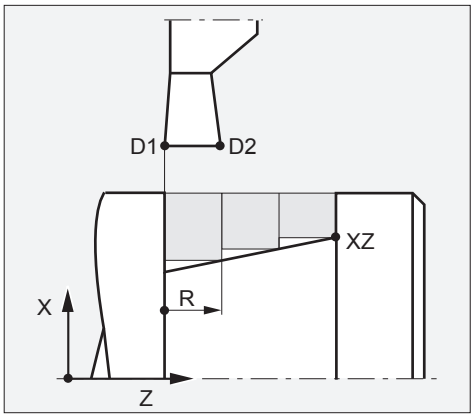



```
%34
N10 G00 G52 X.. Z..                                  Tool change position
N20 T03 D03 M06 (GROOVE ROUGHING TOOL R=0.4)
N30 S900 M03 M40
N40 X52 Z-15                                         Point A, tool at the start of the groove
N45 G92 S3000
N50 G96 S100
N60 G95 F0.2
N70 G65 N100 N180 Z-40 I0.2 K0.1 EA-135 P2
N80 G52 X..Z.. G97 S900
N90 T05 D05 M06 (SLIDING TOOL R=0.8)
N100 G42 X44 Z2                                      Point a, tool at the start of the profile
N110 G96 S250
N120 G01 X50 Z-1 F0.1                                Point b
N130 Z-15                                            Point c
N140 X30 Z-25                                        Point d
N150 Z-35                                            Point e
N160 G02 X40 Z-40 R5                                 Point f
N170 G01 X50                                         Point g
N180 X60 Z-45                                        Point h
N190 G00 G40 G52 X.. Z.. G97 S900 M05
N.. ...
```

5.13.6 Plunging Cycle ► G66

G66	Plunging cycle.

This function is used to rough a dia. or frontal groove by consecutive penetrations.



Syntax

N.. **G66** D.. X.. Z.. [EA..] P../R.. [EP..] [EF..]

G66	Plunging cycle.
D..	Number of the second grooving tool correction (the first correction must be programmed in one of the previous blocks, see Fig. 1).
X.. Z..	Groove machining end point (see Fig. 2).
EA..	Angle defining the slope across the groove bottom (see Fig. 2). - The default value of EA = 0 (no slope).
P../R..	Step between each penetration. P..: value along X (frontal groove). R..: value along Z (dia. groove).
EP..	Retraction (at 45 degrees) at the end of the pass (see example of special use of the cycle).
EF..	Dwell at the end of each penetration in seconds (Format F022). - The default value of EF = 0.

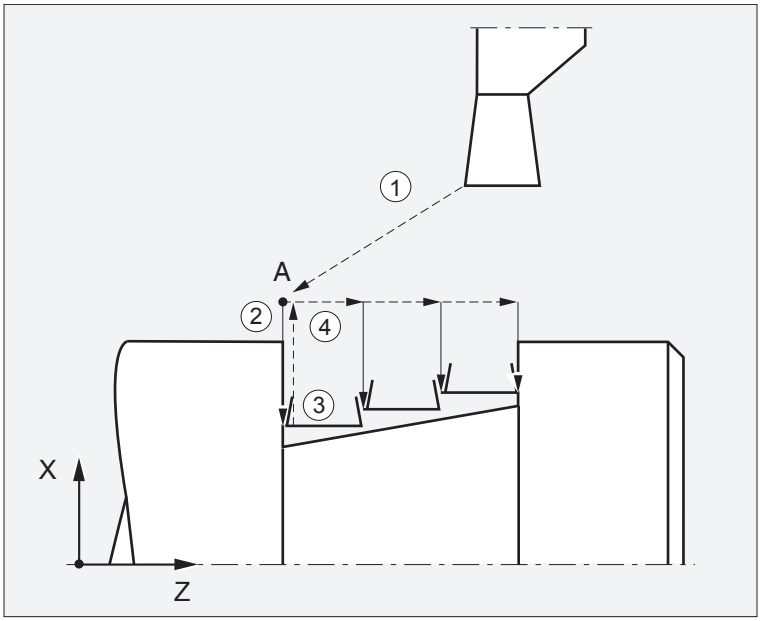
Property of the Function

Function G66 is nonmodal.

Cancellation

Function G66 is cancelled at the end of the block.

Cycle Steps



- Step 1: Tool positioning above the material (point A, cycle start).
- The first tool correction corresponding to the start side of the groove must be programmed in this block or in one of the previous blocks.
- Step 2: Penetration at the machining feed rate
  (to the groove bottom point possibly modified by angle EA).
- Step 3: Retraction at the machining feed rate.
- Step 4: Rapid movement by the value of the step for execution of the next penetration
  (taking into account the correction corresponding to the other side of the groove).

Execution of the following passes in the same way as steps 2 to 4 until reaching the end of the groove.

After execution of the last pass, retraction clear of the material.

Notes

When the cycle is programmed, the system must be in state G40 (G41 or G42 tool radius offset cancelled).

The tool penetrations are uniformly distributed over the width of the groove. In certain cases, the system recomputes the programmed step.

The feed rate function and argument can be programmed in the cycle block, e.g.:

```
N.. G66 D.. X.. Z.. R.. EA.. EF.. G95 F0.1.
```

At the end of the cycle, the system is left in G0 mode.

Declaration of the two tool corrections

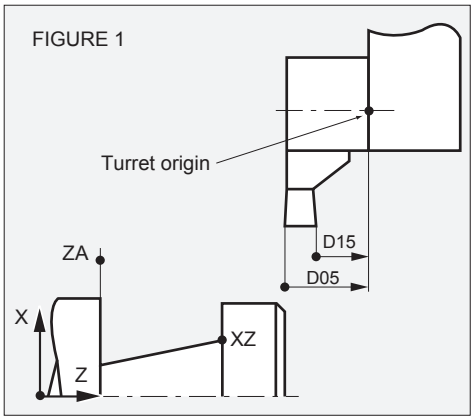
Example:

Correction D05: Groove start side along Z (point A).

Correction D15: Groove end side along Z (point XZ).

The difference in value between the two corrections must be equal to the tool width.

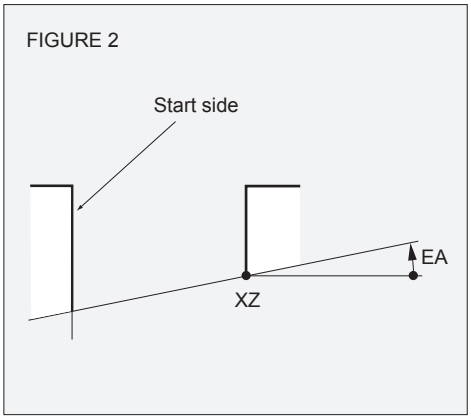
FIGURE 1



Notes on arguments XZ and EA

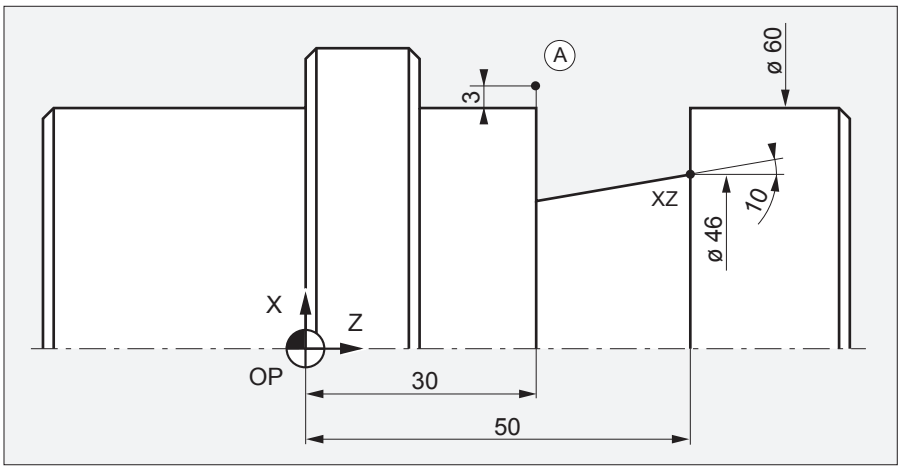
The groove bottom slope is defined from the groove end point (point XZ) to the intersection with the groove start side.

FIGURE 2



Examples

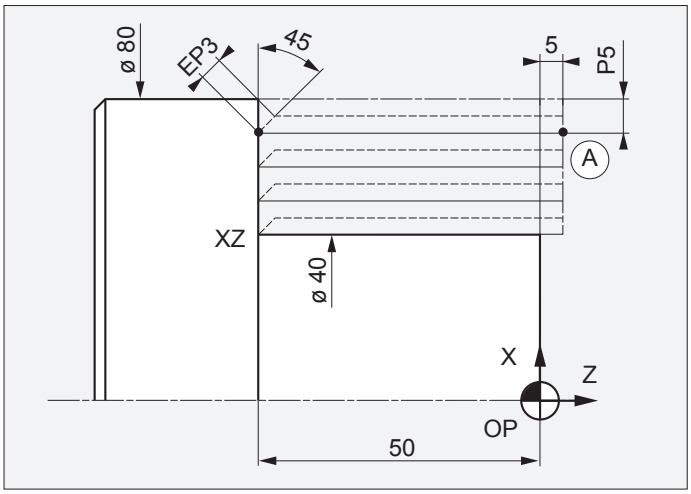
Plunging an axial groove




```
N.. ...
N210 G00 G52 X.. Z..                    Tool change position
N220 T05 D05 M06 (GROOVING TOOL L=6)
N230 G97 S900 M40 M03
N240 X66 Z30                            Point A
N250 G96 S80
N260 G95 F0.1
N270 G66 D15 X46 Z50 EA10 R5 EF0.5      Cycle
N280 G52 X.. Z.. G97 S900 M05
N..
```

Special use of cycle G66 with EP

EP can be added to the cycle syntax to rough a diameter.



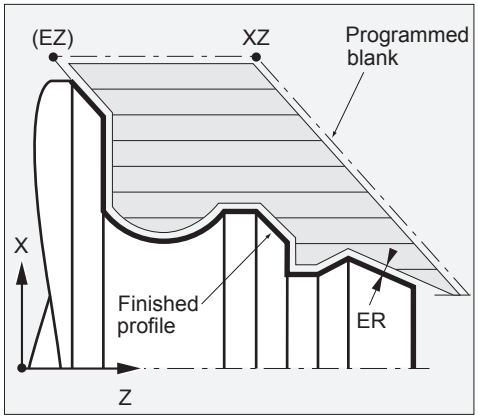

```
N.. ...
N100 G00 G52 X.. Z..                    Tool change position
$0 TURNING, DIAMETER 40
N110 T01 D01 M06 (BIDIRECTIONAL TOOL R=08)
N120 G97 S900 M40 M03
N130 X70 Z5                             Point A
N140 G96 S200
N150 G95 F0.3
N160 G66 X40 Z-50 P5 EP3                Cycle
N170 ...
```

5.13.7 Roughing Cycle with Groove ► G63

G63	Roughing cycle with groove.

This function is used for roughing a volume of material located between the definition of a blank and a finished profile.

The cycle will machine a groove whose profile is compatible with the geometry and radius of the tool used. It can be carried out by facing or turning and for external or internal machining.



Syntax

N.. **G63** [N.. N..] / X.. Z.. EX.. / EZ.. P.. / R.. EA.. / EU.. / EW.. [EB..] [EC..] [ER..] [Q..] [EQ..] [EF..]

G63	Roughing cycle with groove.
N.. N..	First and last block numbers defining the finished profile (maximum 95 blocks).
X.. Z.. (or U.. W..)	Cycle start point.
EZ.. / EX..	Pass end point on the roughing axis: - EZ for rough turning along Z. - EX for rough facing along X. Direction of roughing pass execution: - Z+ if EZ > Z Z- if EZ < Z - X+ if EX > X X- if EX > Z
P.. / R..	Pass depth: - P: value along X (rough turning) - R: value along Z (rough facing).
EA../EU../EW..	Roughing pass start positions (see Figs. 1 and 2). These arguments are programmed to define a start angle for cut application and can be combined: EA EU, EA EW or EU EW.
EB..	Maximum groove penetration angle. Angle used to take into account the groove profile geometry and the tool rake angle (α) (see Fig. 3). - By default, EB is parallel to the roughing axis.
EC..	Maximum cut exit angle. Angle used to take into account the groove profile geometry and the tool angle of attack (K) (see Fig. 4). - By default, EC is perpendicular to the roughing axis.
ER..	Finishing allowance. Value applied normal to the profile. - The default value of ER = 0.
Q..	Positioning clearance. Approach distance at feed rate before contact with the material (for rapid to next cut depth). - The default value of Q = 0 (no clearance).
EQ..	Minimum chip value. - Below the programmed value, the pass is not executed.
EF..	Penetration feed rate in the material. - The default value of EF is the same as the modal feed rate F.

Property of the Function

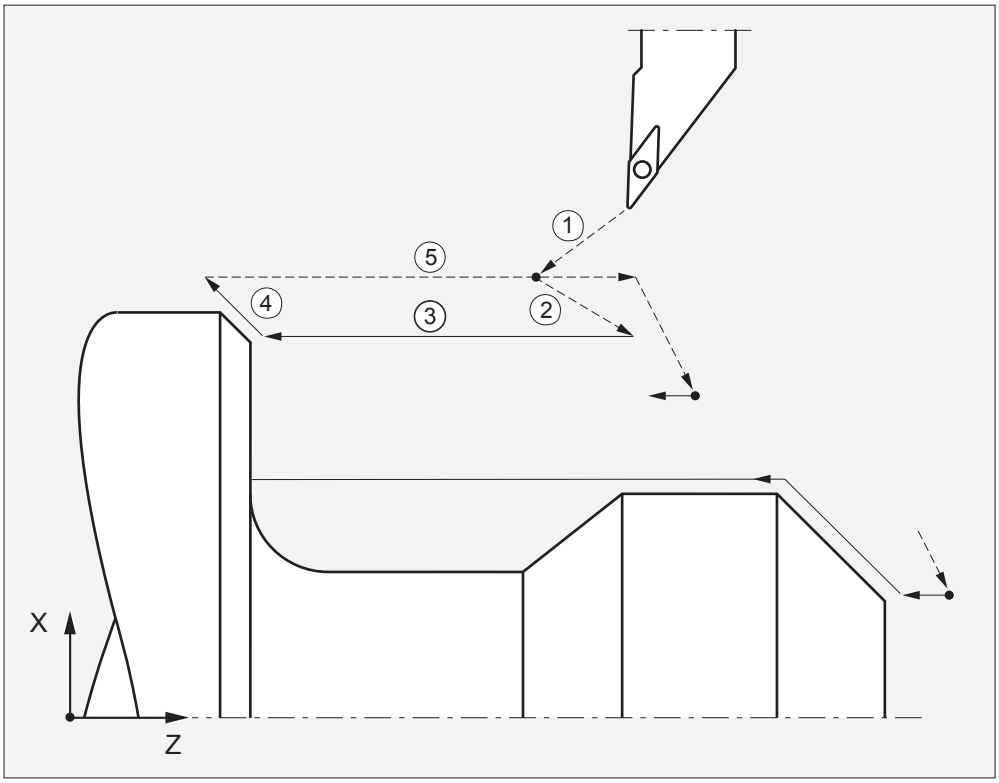
Function G63 is nonmodal.

Cancellation

Function G63 is cancelled at the end of the block.

Cycle Steps

Description of the first rough turning and grooving passes

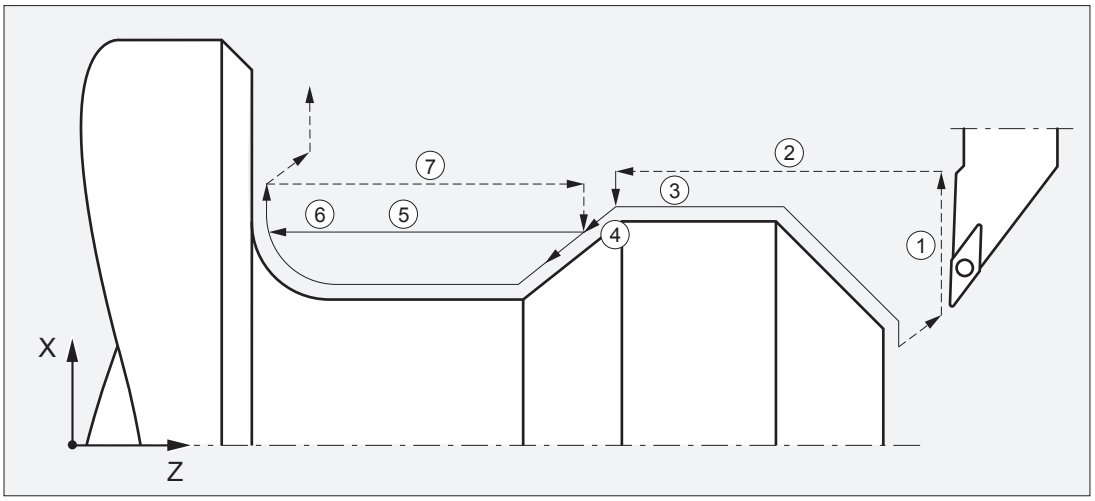


Rough turning pass

- Step 1: Tool positioning to X Z.
- Before the cycle begins, the system reads the declared tool radius.
- Step 2: Rapid to first cut depth P (for turning).
- Step 3: Execution of the first pass at the machining feed rate.
- Step 4: Lift along the profile at the machining feed rate.
- Step 5: Rapid return to the cycle start point.

The following passes are executed in the same way as steps 2 to 5.

After execution of the last roughing pass, the tool is retracted from the surface and positioned in X.



Grooving pass

- Step 1: Rapid positioning in X.
- Step 2: Rapid positioning in Z.
- Step 3: Positioning in X at the machining feed rate.
- Step 4: Cut application at the machining feed rate along the groove profile.
- Step 5: Execution of the pass at the machining feed rate.
- Step 6: Lift at the machining feed rate along the groove profile.
- Step 7: Rapid return to the cycle start point.

The following passes are executed in the same way as steps 3 to 6.

After execution of the last pass, the tool is retracted from the surface and positioned in X.

Notes

The number of blocks from N.. to N.. must not exceed 95. Otherwise, the system returns error message 92.

When the cycle is programmed, the system must be in state G40 (G41 or G42 tool radius offset cancelled).

At the end of the cycle, the system is left in G0 mode.

The finished profile definition can be programmed before the cycle call, but in this case it is necessary to use the branch function G79 (see chapter 5.14.3).

The cycle arguments defining positive angles are specified counterclockwise.

The feed rate function and argument can be programmed in the cycle block, e.g.:

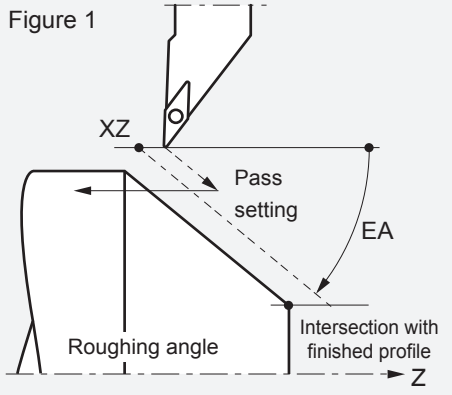
```
N.. G63 N.. N.. X.. Z.. EZ.. P.. EA.. EB.. ER.. Q.. EQ.. EF.. G95 F0.3
```

Notes on arguments EA, EU and EW

- EA: Angle limiting the start of cut application between point XZ (cycle start) and intersection with the finished profile.
  (If there is no intersection, the last pass is limited to the first point of the finished profile)

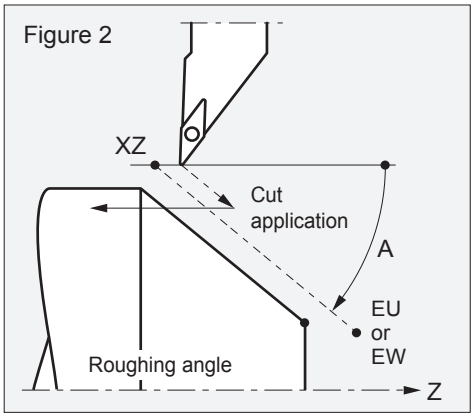
After positioning to point XZ, the system reads the declared tool radius before carrying out the first cut application.

Figure 1



- EU or EW: End point of the last roughing pass. If argument EA is not programmed, the values declared with EU and/or EW and XZ (cycle start) are used to define the cut application start angle (A).

Figure 2



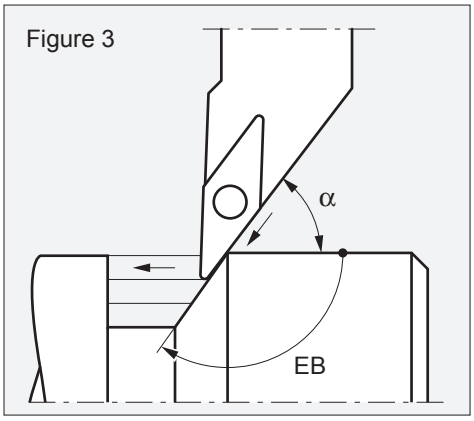
The angle is also used to differentiate between applying the cut at feedrate or rapid for a groove or not, respectively:

- Machining feed rate (in the material) if the angle between the cut application line and the roughing axis is $\geq$ 90 degrees.
- Traversing rate (off the material) if the angle between the cut application line and the roughing axis is < 90 degrees.

5.13.7.1 Definition of EB and EC

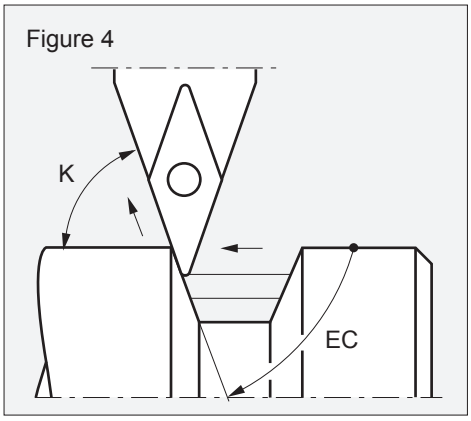
Angle defined by EB

Figure 3



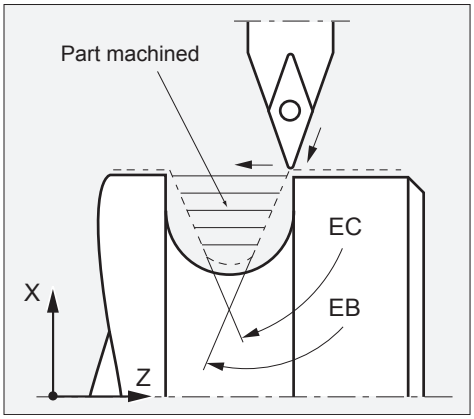
Angle defined by EC

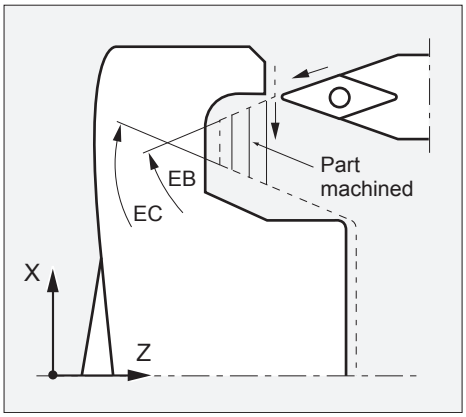
Figure 4



Machining of a groove limited by declaration of angles EB and EC

When the groove profile and tool geometry are completely incompatible, the groove is not executed.

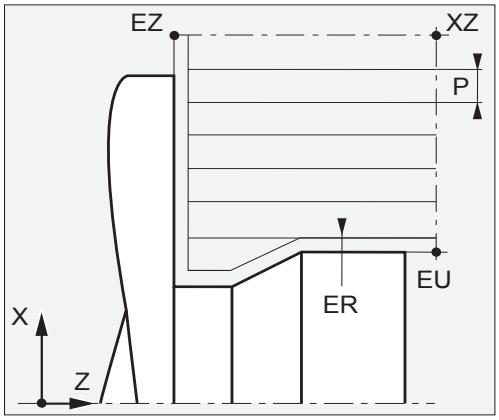




5.13.7.2 Cycle differences between turning and facing

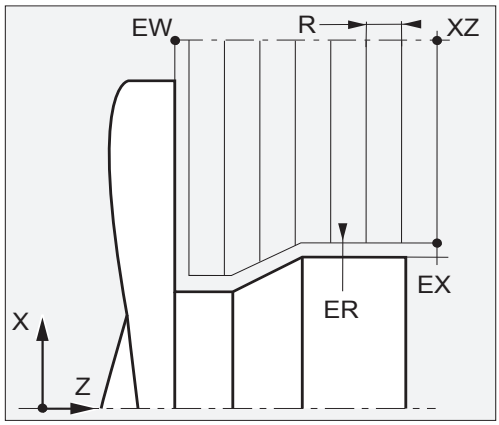
Rough turning

```
N70 ...
N80 G63 N.. N.. X.. Z.. EZ.. P.. EU.. EB.. ER..
N90 G52 X.. Z..
N..
```



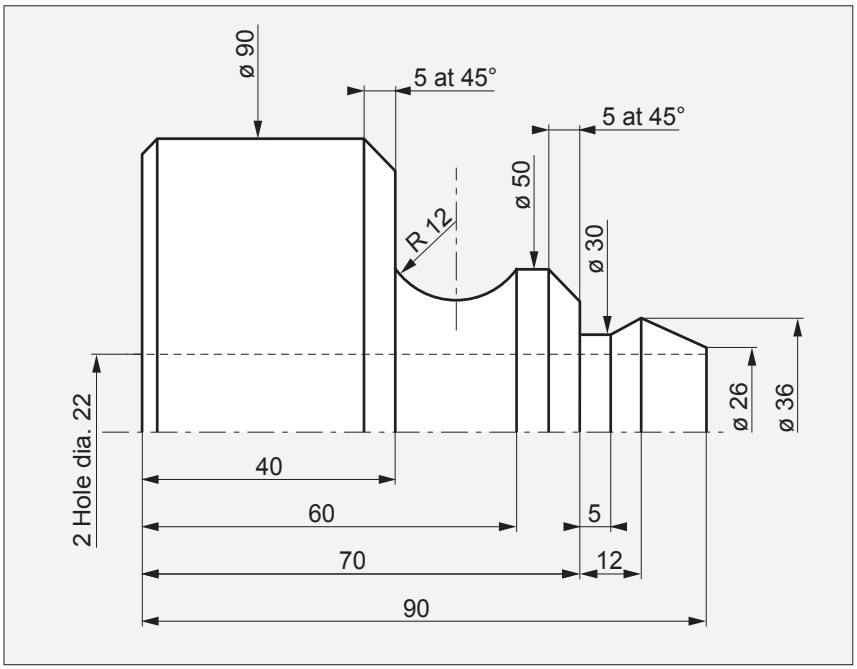
Rough facing

```
N70 ...
N80 G63 N.. N.. X.. Z.. EX.. R.. EW.. EC.. ER..
N90 G52 X.. Z..
N..
```

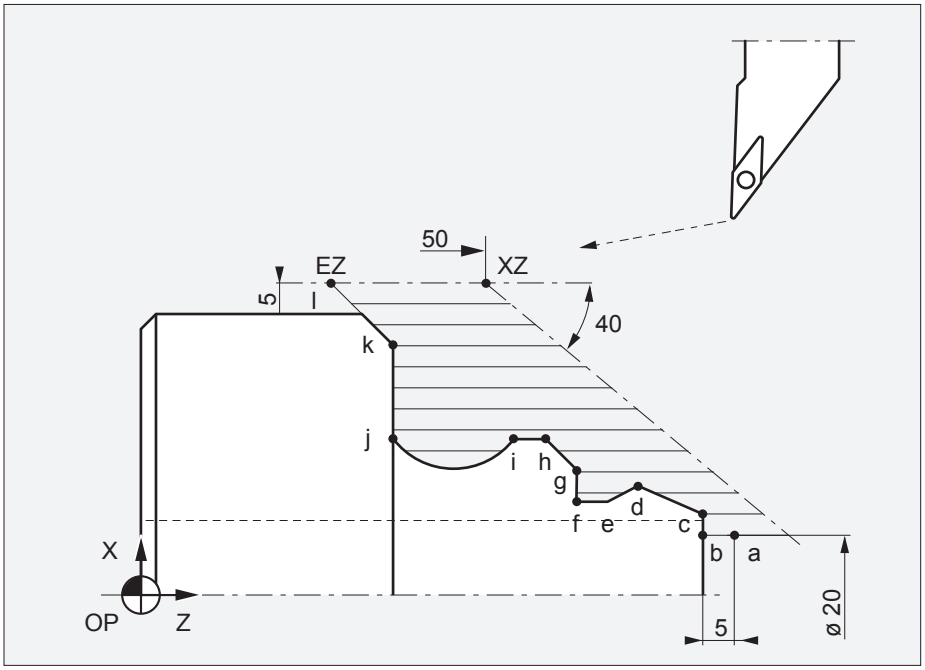


Examples

Rough and finish turning of an external profile



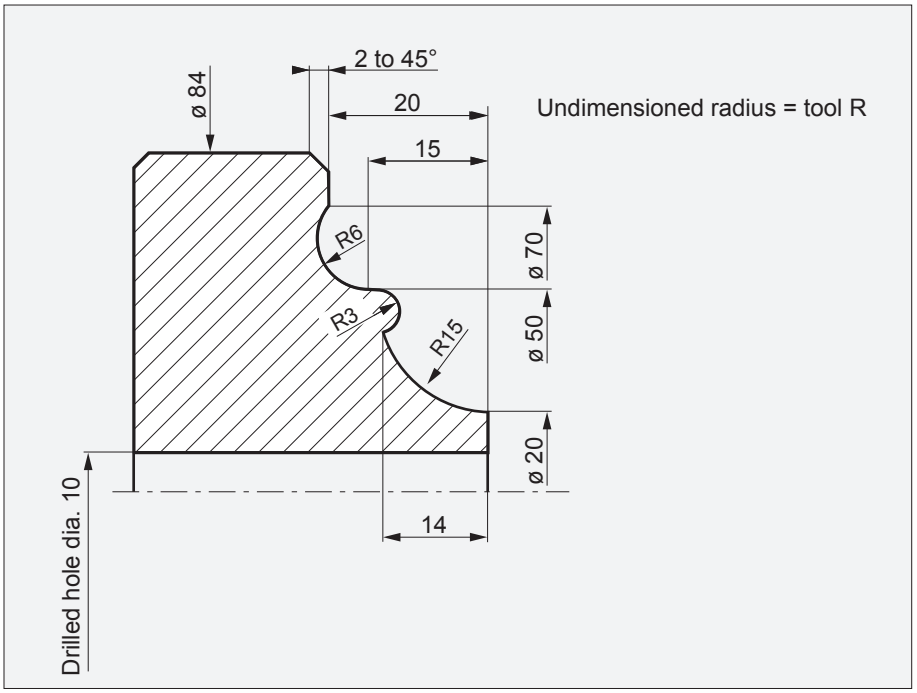
Machining paths



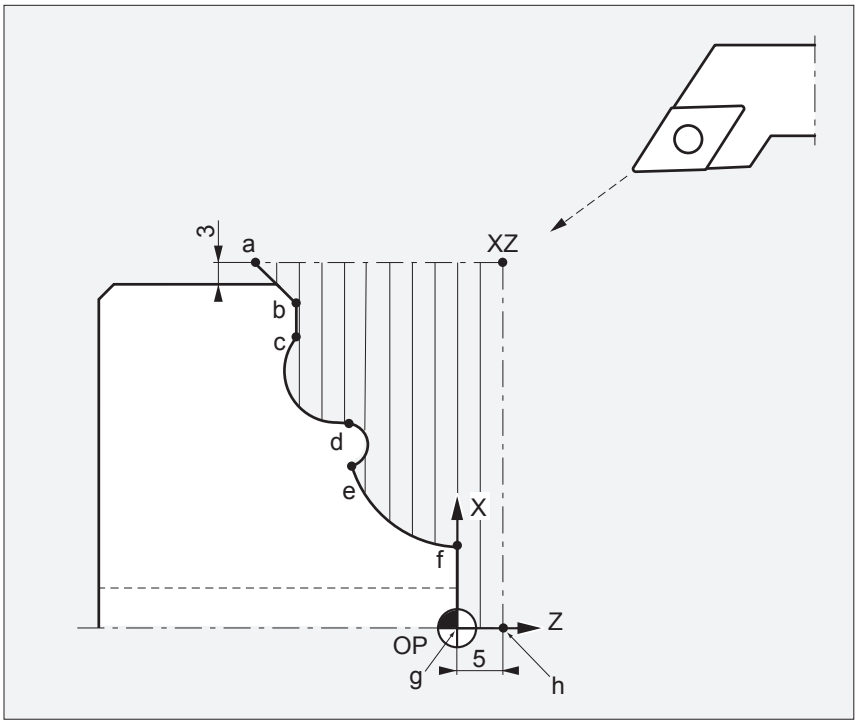

```
%32
N10 G00 G52 X.. Z..                          Tool change position
N20 T01 D01 M06 (SLIDING TOOL R=0.8)
N30 S900 M40 M04
N40 X100 Z50                                 Tool position at the start of roughing
N50 G92 S3500
N60 G96 S200
```

```
N70 G95 F0.25
N80 G63 N110 N230 X100 Z50 EZ30 P2 EA-40 EB-145 ER0.4 Q2 EQ1 EF0.1
N90 G52 X.. Z..
N100 T03 D03 M06 (COPYING TOOL R=0.4)
N110 G42 X20 Z95                    Point a, tool at the start of finishing
N120 G96 S250
N130 G01 Z90 F0.4                   Point b
N140 X26 F0.1                       Point c
N150 X36 Z82                        Point d
N160 X30 Z75                        Point e
N170 Z70                            Point f
N180 X40                            Point g
N190 X50 Z65                        Point h
N200 Z60                            Point i
N210 G02 X50 Z40 R12                Point j
N220 G01 X80                        Point k
N230 X100 Z30                       Point l
N240 G40 G52 X.. Z.. G97 S900 M05
N250 M02
```

Rough and finish facing of an external profile



Machining paths

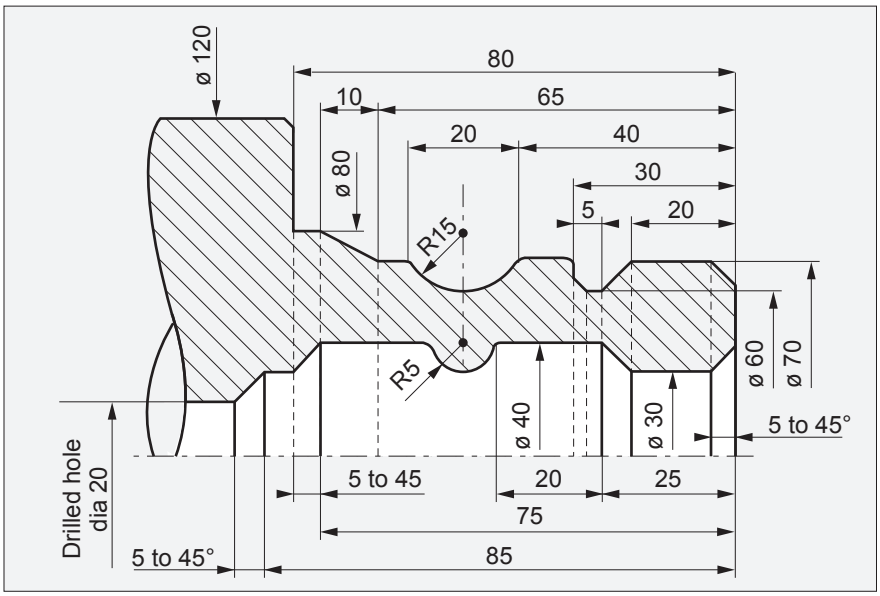



```
%39
N10 G00 G52 X.. Z..                          Tool change position
N20 T07 D07 M06 (FACING TOOL =0.8)
N30 S900 M40 M04
N40 X100 Z10                                 Tool position at the start of roughing
N50 G92 S3500
N60 G96 S200
N70 G95 F0.2
N80 G63 N110 N190 X90 Z5 EX0 R2 EA-180 EB-150 EC-170 ER0.2 Q1 EF0.1
```

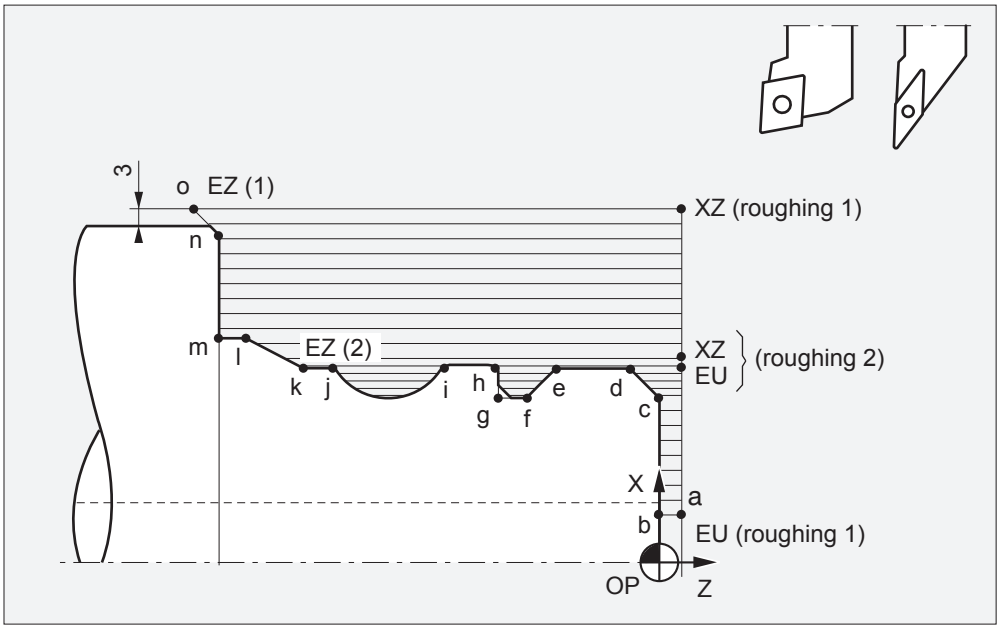
```
N90 G52 X.. Z..
N100 T09 D09 M06 (COPYING TOOL R=0.4)
N110 G41 X90 Z-25                    Point a, tool at the start of finishing
N120 G96 S250
N130 G01 X80 Z-20 F0.1               Point b
N140 X70                             Point c
N150 G03 X50 Z-15 R6                 Point d
N160 G02 X40 Z-14 R3                 Point e
N170 G03 X20 Z0 R15                  Point f
N180 G01 X0                          Point g
N190 G00 Z5                          Point h
N200 G40 G52 X.. Z.. G97 S900 M05
N210 ...
N.. M02
```

Execution of a profile by two outside rough turnings then finishing

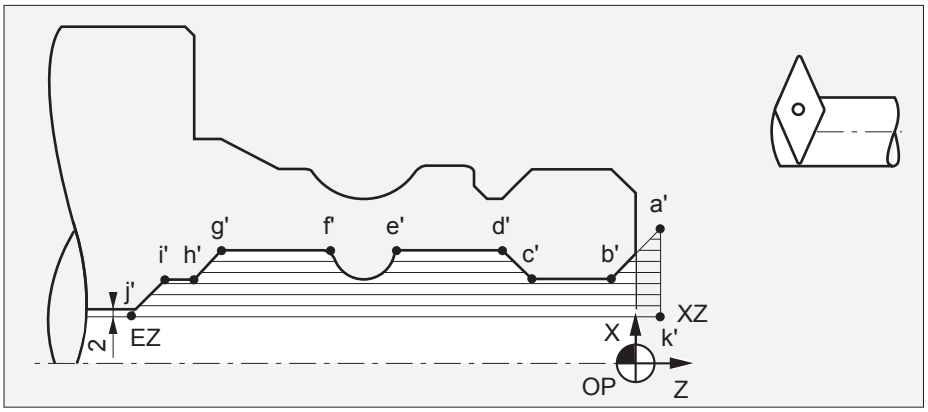
Rough and finish turning an internal profile



External machining paths



Internal machining paths




```
%163
N10 G0 G52 X.. Z..                          Tool change position
$0 EXTERNAL ROUGHING 1
N20 T01 D01 M06 (TOOL R0.8)
N30 S900 M40 M04
N40 X126 Z5                                 Tool position at the start of roughing 1
N50 G92 S4000
N60 G96 S260
N70 G95 F0.3
N80 G63 N150 N300 X126 Z5 P3 EZ-85 EU16 EB-170 EC-93 ER0.2 EQ1 Q1 EF0.2
N90 G0 G52 X.. Z.. G97 S900
$0 EXTERNAL ROUGHING 2
N100 T03 D03 M06 (TOOL R0.8)
N110 X80 Z5                                 Tool position at the start of roughing 2
N120 G63 N150 N300 X72 Z5 P1.5 EZ-65 EU70 EB-125 EC-93 ER0.2 Q1 EF0.15
N130 G0 G52 X.. Z.. G97 S900
$0 EXTERNAL FINISHING
N140 T11 D11 M06 (COPYING TOOL R0.8)
N150 G42 G0 X16 Z5                          Point a
N160 G96 S300
N170 G01 Z0 F0.08                           Point b
N180 X60                                    Point c
N190 X70 Z-5                                Point d
N200 Z-20                                   Point e
N210 X60 Z-25                               Point f
N220 Z-30 EB-2                              Point g
N230 X70 EB2                                Point h
N240 Z-40 EB2                               Point i
N250 G02 X70 Z-60 R15 EB2                   Point j
N260 G01 Z-65                               Point k
N270 X80 Z-75                               Point l
N280 Z-80                                   Point m
N290 X116                                   Point n
N300 X126 Z-85                              Point o
N310 G0 G40 G52 X.. Z.. G97 S900 M05
$0 INTERNAL ROUGHING
N320 T05 D05 M06 (BORING TOOL R0.8)
N330 S900 M40 M03
N340 X10 Z10                                Tool position at the start of roughing
N350 G92 S3000
N360 G96 S70
N370 G95 F0.1
N380 G63 N420 N520 X16 Z5 EU50 EZ-92 P2 EB110 EC70 ER0.2 Q1 EQ0.5 EF0.1
N390 G0 Z5                                  Tool retraction
```

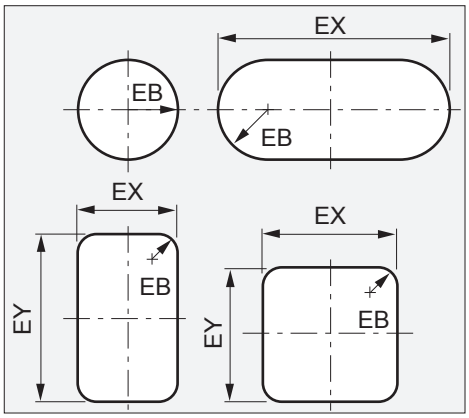
```
N400 G0 G52 X.. Z..
$0 INTERNAL FINISHING
N410 T07 D07 M06 (BORING TOOL R0.4)
N420 G41 X50 Z5                          Point a'
N430 G96 S90
N440 G01 X30 Z-5 F0.07                   Point b'
N450 Z-20                                Point c'
N460 X40 Z-25                            Point d'
N470 Z-45 EB2                            Point e'
N480 G02 X40 Z-55 I40 K-50 EB2           Point f'
N490 G01 Z-75                            Point g'
N500 X30 Z-80                            Point h'
N510 Z-85                                Point i'
N520 X16 Z-92                            Point j'
N530 G0 Z5                               Point k', Tool retraction
N540 G40 G52 X.. Z.. G97 S900 M05
N550 M02
```

5.14 Simple Pocket Cycle ► G45

G45	Simple pocket cycle.

This cycle is used to machine circular, oblong, rectangular and square pockets.

The primary and secondary axes are programmable in absolute dimensions and define the pocket center in the plane and the pocket depth in the tool axis.



Syntax (XY plane)

N.. [G17] **G45** X.. Y.. Z.. [ER..] EX.. EY.. [EB..] P.. Q.. [I..] [J..] [EG2/EG3] EP.. EQ.. EI.. EJ..

G17	XY plane selection.
G45	Pocket cycle.
X.. Y..	Position of the pocket center.
Z..	Pocket bottom end point
ER..	Retraction plane in the tool axis.
EX..	Pocket dimension on the X or U axis.
EY..	Pocket dimension on the Y or V axis.
EB..	Radius of a circular pocket if EB is programmed alone. Radius of an oblong pocket. Fillet for other pockets.
P..	Axial roughing path.
Q..	Lateral roughing path.
I..	Axial finishing path.
J..	Lateral finishing path.
EG2/EG3	Pocket cutting direction (default EG3) - EG2: Clockwise (Conventional milling) - EG3: Counterclockwise (Climb milling).
EP..	Value of the axial roughing feed rate.
EQ..	Value of the lateral roughing feed rate.
EI..	Value of the axial finishing feed rate.
EJ..	Value of the lateral finishing feed rate.

Properties of the Function

Function G45 is nonmodal. None of the arguments in the cycle are modal.

Cancellation

Function G45 is cancelled at the end of the block.

Notes

When the start of spindle rotation is programmed in the block, it must be placed before function G45 and its arguments.

Example: N.. S1000 M03 M40 G45...

If tool correction D.. is missing when cycle G45 is called, the system returns error message 898.

When the cycle is programmed, the system must be in state G40 (G41 or G42 radius offset cancelled).

When executing a cycle programmed with axial or lateral roughing and/or finishing passes, if only one feed rate (EP, EQ, EI or EJ) is programmed, this feed rate is used as default.

If no:

- axial feed rate (roughing or finishing) is programmed, the system returns error message 892,
- lateral feed rate (roughing or finishing) is programmed, the system returns error message 893.

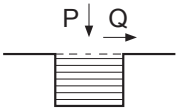
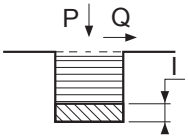
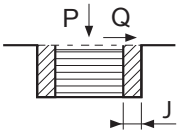
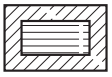
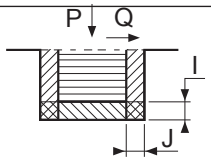
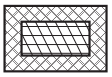
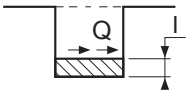
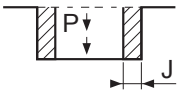
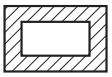
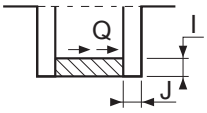
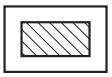
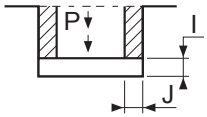
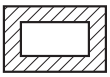
The cycle calls the axis address equivalence table (see chapter 7.4). Use of the table is incompatible with the cycle call (it is initialised at the beginning of the cycle and restored initialised at the end of the cycle).

In the ZX and YZ planes, the pocket dimension in the Z (or W) axis is programmed by EZ..

During machining:

- The cycle cannot be interrupted until the pocket cycle is completed on a depth level (no possibility of mode change).
- The cycle cannot be modified until it has been completely executed.

5.14.1 Pass setting possibilities

Lateral, axial roughing	Lateral finishing	Axial finishing	SIDE VIEW	TOP VIEW
P . . Q . . Axial and lateral roughing				
P . . Q . . I . . Axial and lateral roughing and axial finishing of the bottom				
P . . Q . . J . . Axial and lateral roughing and lateral finishing of the sides				
P . . Q . . I . . J . . Sequenced axial and lateral roughing and lateral finishing (for each axial pass setting)				
Q . . I . . Axial finishing of the bottom by value I				
P . . J . . Lateral finishing of the sides by value J				
Q . . I . . J . . Axial and lateral finishing of the bottom up to lateral value J				
P . . I . . J . . Lateral finishing of the sides up to axial value I				

5.14.2 Roughing cycle with a single tool

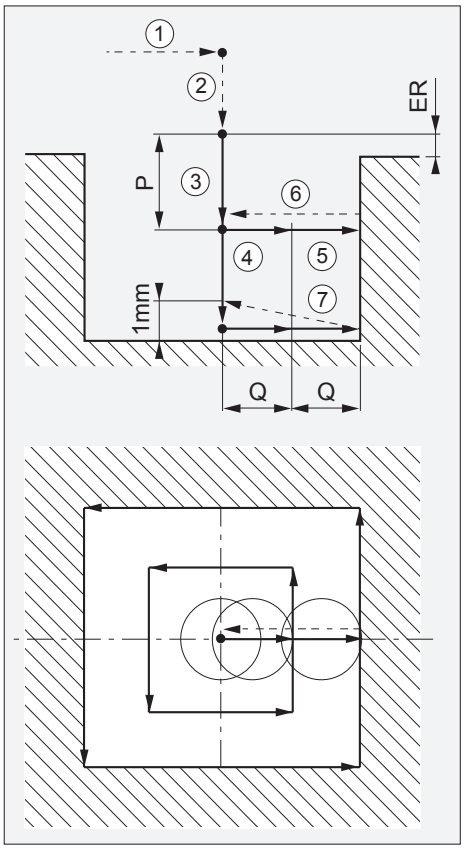
Cycle with axial (P) and lateral (Q) roughing parameters.

Example:

```
G45 X.. Y.. Z.. ER.. EX.. EY.. P.. Q.. EP.. EQ..
```

5.14.3 Roughing Cycle Steps

- Step 1: Rapid positioning of the tool to the center of the pocket.
- Step 2: Rapid axial positioning in the tool axis.
- Step 3: Axial penetration at the programmed feed rate to depth P.
- Step 4: Lateral positioning at the programmed feed rate to value Q (along the small side).
- Execution of the first pocket contouring (and subsequent contourings if any).
- Step 5: Lateral positioning on the final contour by value Q.
- Execution of the last contouring to the external pocket dimensions.
- Step 6: Rapid repositioning to the pocket center for execution of a new plunge and a new contouring to depth P following the same steps as 3, 4, 5.
- Step 7: After execution of the last contouring to depth Z, the tool is rapidly repositioned to the pocket center with a clearance of 1 mm in the tool axis then retracted to programmed position ER.

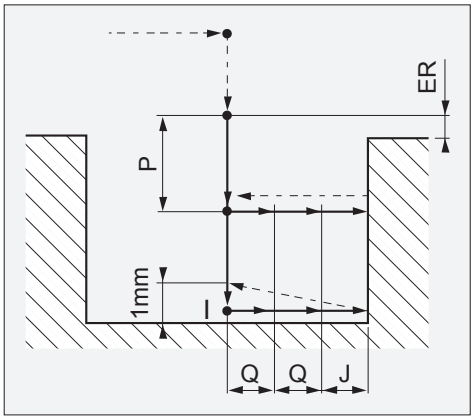


5.14.4 Roughing and finishing cycle with a single tool

Cycle with axial (P) and lateral (Q) roughing and axial (I) and lateral (J) finishing parameters.

Example:

```
G45 X.. Y.. Z.. ER.. EX.. EY..
P.. Q.. I.. J.. EP.. EQ.. EI.. EJ..
```

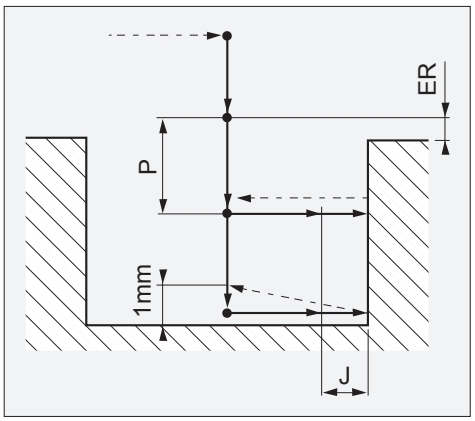


5.14.5 Finishing cycle alone with a new tool

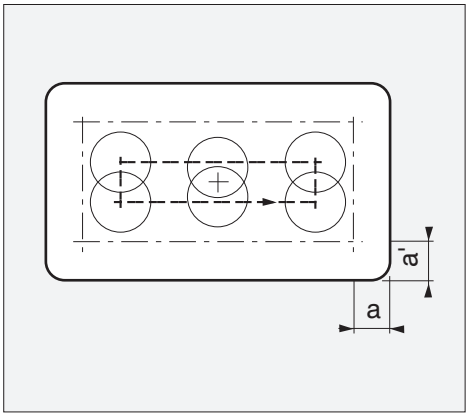
Cycle with depth (P) and lateral finishing (J).

Example:

```
G45 X.. Y.. Z.. ER.. EX.. EY..
P.. J.. EP.. EJ..
```



Oblong, square and rectangular pockets are roughed in such a way that the values of the lateral passes in the two axes are equal (a = a').



5.14.6 Pass setting in the plane (Q)

R=Tool radius, L=dimension of the small side

- If Q > 2R, the system returns error message 887.
- If Q = or < R and the result of division of the half-distance to be roughed by the pass set is not an integer, the system computes a new setting for pass Q.

The number of passes is equal to (L/2-R)/Q = m (rounded to the integer above if necessary).

Determination of the new pass setting: Q' = (L/2-R)/n

The value of Q' is rounded off to the next higher micrometer (µm).

In this case, the value of the last pass is:

- L/2-R-(n-1)Q' (value less than Q').
- If R < Q < 2R, the value of the first pass is R/2 (except for circular pockets) and the value of the subsequent passes is Q'=L/2-R/n.

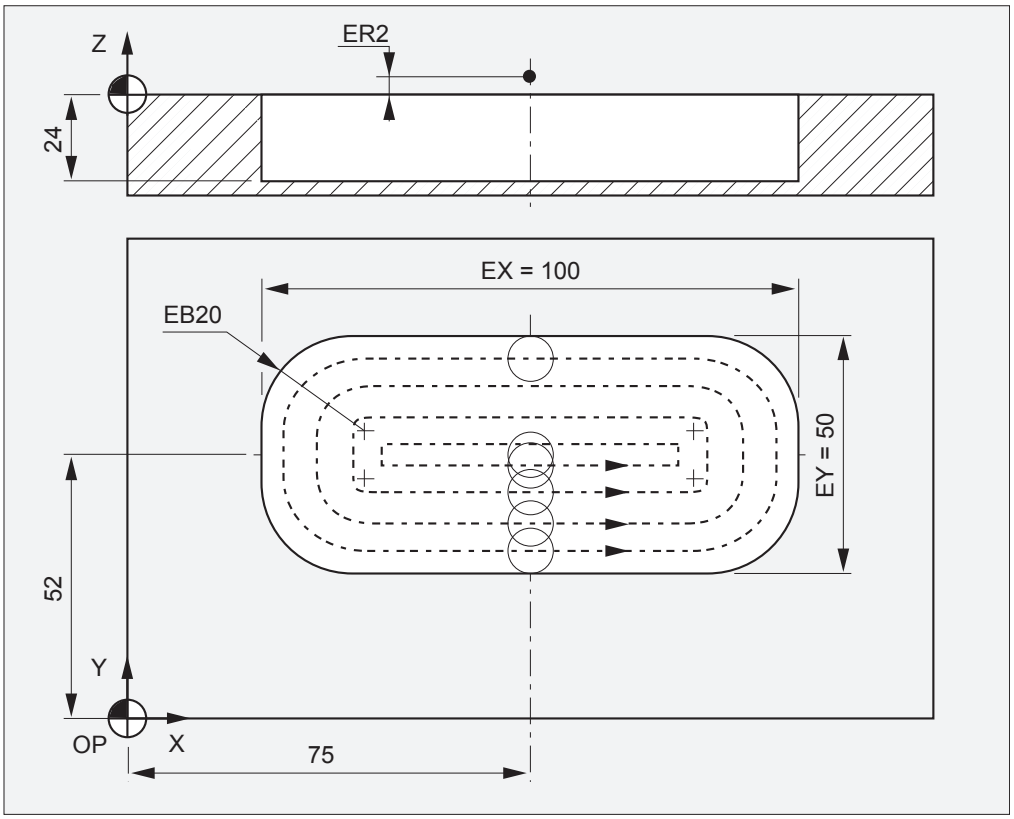
If the cutter radius does not allow at least one roughing pass or the finishing pass to be cut at the programmed value, the system returns error message 896.

5.14.6.1 Pass setting in the tool axis (P)

The system computes a new pass setting P' using the same formula as for Q' if the result of division of the total depth by the pass set is not an integer.

Examples

Execution of a rectangular pocket in the XY plane (G17)



Roughing alone without finishing.

```
%55
N10 G00 G52 Z..
N20 T04 D04 M06 (CUTTER DIAMETER=8 CENTER CUT)
N30 S3000 M03 M40
N40 G45 X75 Y52 Z-24 ER2 EX100 EY50 EB20 P5 Q7 EP100 EQ500
N50 G00 G52 Z..
N..
```

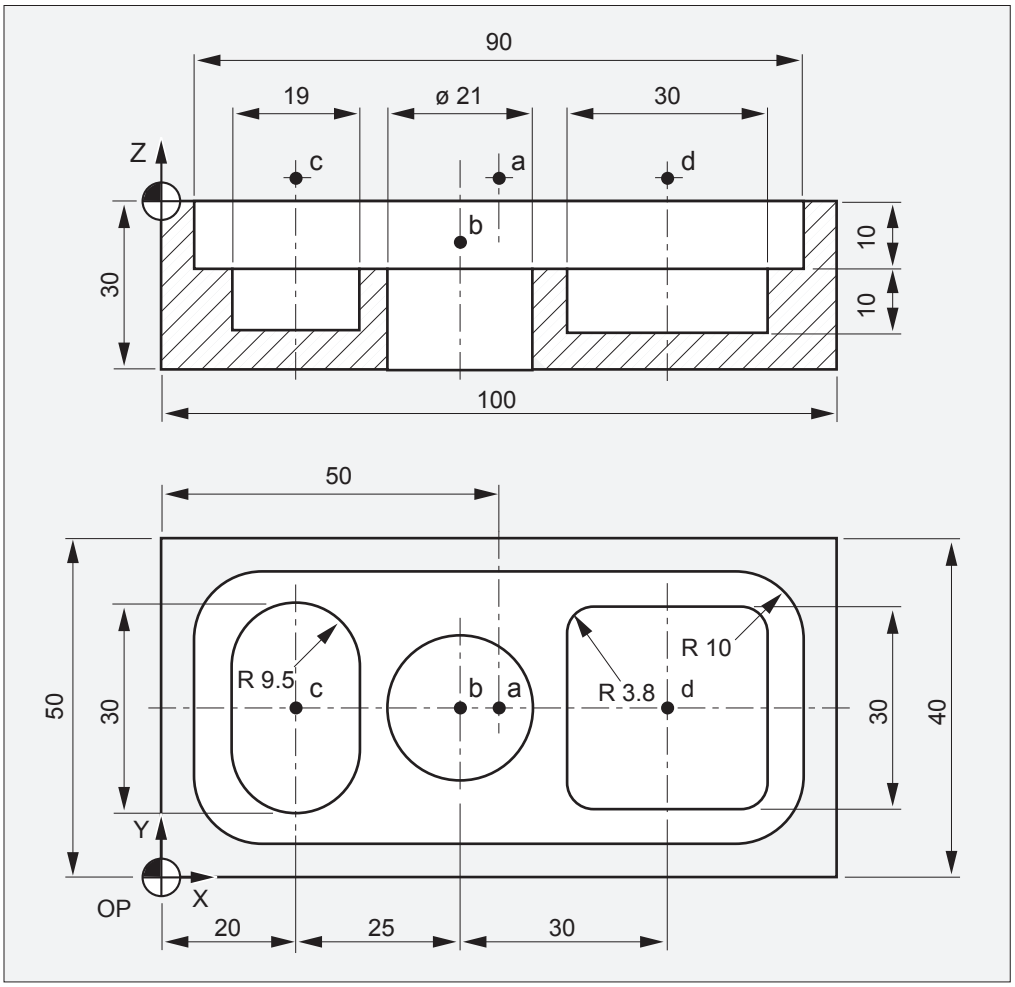
Roughing and finishing with the same tool.

```
%55
N10 G00 G52 Z..
N20 T04 D04 M06 (CUTTER DIAMETER=8 CENTER CUT)
N30 S3000 M03 M40
N40 G45 X75 Y52 Z-24 ER2 EX100 EY50 EB20 P5 Q7 I0.3 J0.3 EP100 EQ500 EI70 EJ300
N50 G00 G52 Z..
N..
```

Execution of a sequence of pockets in the XY plane

The rectangular, circular and oblong pockets are executed with a single tool.

The square pocket is executed with two tools (roughing + finishing tool).



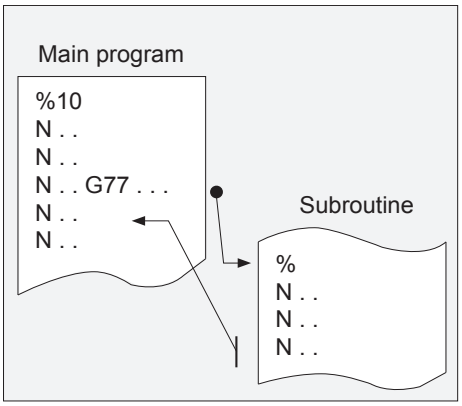

```
%206
N10 G00 G52 Z..
N20 T01 D01 M06 (CUTTER DIAMETER=10 CENTER CUT)
N30 S600 M40 M03 M08
$0 RECTANGULAR POCKET
N40 G45 X50 Y25 Z-10 ER2 EX90 EY40 EB10 P3 Q8 EP50 EQ150
$0 CIRCULAR POCKET
N50 G45 X45 Y25 Z-26 ER-8 EB10.5 P3 Q8 EP50 EQ150 J0.5 EJ200
$0 OBLONG POCKET
N60 G45 X20 Y25 Z-20 ER-8 EY30 EB9.5 P3 Q8 EP50 EQ150 I0.5 J0.5 EI70 EJ200
$0 SQUARE POCKET ROUGHING
N70 G45 X75 Y25 Z-20 ER-8 EX29.6 EY29.6 P3 Q8 EP50 EQ150
N80 G00 Z200 M05 M09
$0 SQUARE POCKET FINISHING
N90 T02 D02 M06 (CUTTER DIAMETER=6 CENTER CUT)
N100 S1000 M40 M03 M08
N110 G45 X75 Y25 Z-20 ER-8 EX30 EY30 EB3.8 P9.5 I0.5 J0.5 EP50 EI50 EJ100
N120 G00 G52 Z.. M05 M09
N130 M02
```

5.15 Breaks in Sequence ► G77

5.15.1 Unconditional Branch to a Subroutine or Sequence of Blocks with Return

G77	Unconditional branch to a subroutine or sequence of blocks with return.

Subroutines included in the main program or separate from it are called by addresses H.. and/or N.. N.. associated with the function.



Syntax

N.. **G77** [H..] [N.. N../N..] [S..]

G77	Unconditional branch to a subroutine or sequence of blocks with return (maximum 8 nesting levels).
H..	Number of a subroutine external to the main program.
N.. N../N..	Number of the first and last block called (If both N.. have the same number or if only one block is programmed, only one block is called).
S..	Number of times the subroutine or sequence of blocks is to be repeated (default = 1, maximum 99 repetitions).

Property of the Function

Function G77 is nonmodal.

Cancellation

Function G77 is cancelled at the end of the block.

Notes

A subroutine called by addresses N.. N.. can be placed after M02.

If argument S is programmed in a block containing other instructions, it must immediately follow the subroutine branch.

If the subroutine branch is defined by two sequence numbers in reverse order (e.g. G77 N200 N10), the system executes the program in the normal sequence from N10 to N200 and no error is displayed.

Context

The calling program context can be saved at the beginning of the called program.

It is then restored at the end of execution of the called program. The programmed status access symbols are used for save and restore.

Inhibiting display of subroutines being executed

Display of the programming context of a subroutine and its internal subroutines during execution can be inhibited. Placing the character «:» after the subroutine number inhibits display. Only the subroutine call block is displayed.

Example

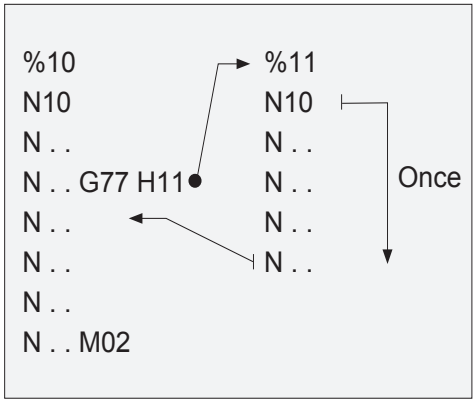
Main program %10 calling subroutine %110: itself containing an internal subroutine %210. Only block N50 of program %10 is displayed during execution of subroutines %110 and %210.

```
%10                  %110:                %210
N10                  N10                  N10
N..                  N..                  N..
N50 G77 H110         N80 G77 H210         N..
N..                  N..                  N..
```

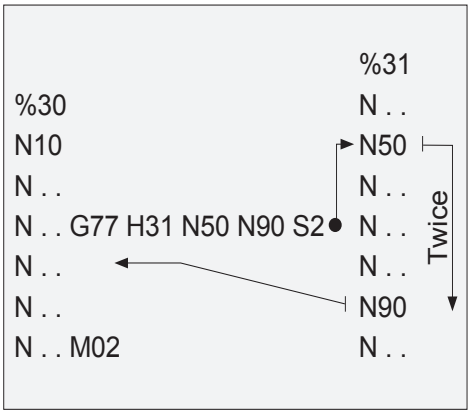
Examples

Branches to external subroutines from the main program

Branch to subroutine %11 from main program %10.

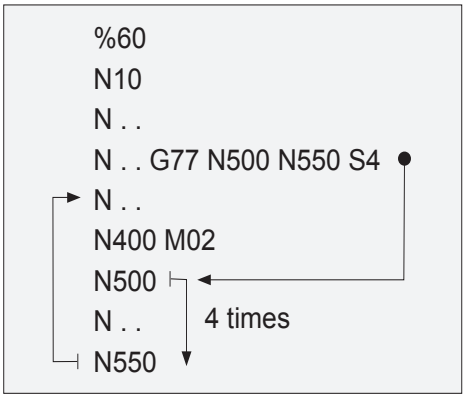


Branch to blocks N50 to N90 of subroutine %31 from main program %30 twice.



Branches to sequences in the program

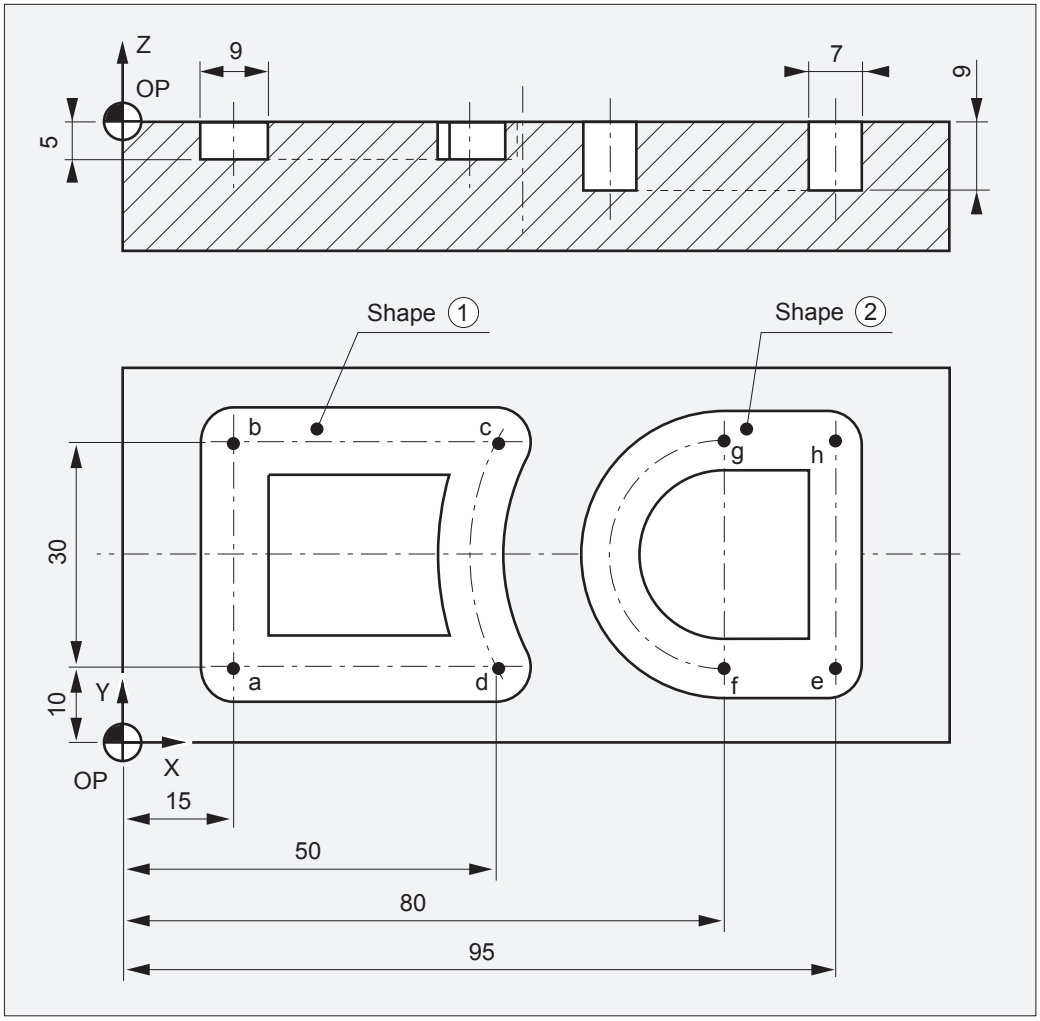
Branch to blocks N500-N550 located downstream in program %60, after M02 four times.



Execution of two grooves by subroutine branches

Shape 1, subroutine H35: two branches for execution of two passes each with depth 2.5 mm.

Shape 2, subroutine H40: three branches for execution of three passes each with depth 3 mm.



Main program and Subroutine for shape 1 and 2

```
%30
N10 G00 G52 Z.. M05 M09
$0 2 SHAPES WITH S/PROG %35 ET %40
N20 T01 D01 M06 (CUTTER DIAMETER=9)
N30 S800 M40 M03
N40 G00 X15 Y10 Z3                    Point a, approach
N50 G01 Z0 F50                        Approach on Z
$0 MACHINING SUBROUTINE %35
N60 G77 H35 S2                        Branch to subroutine %35
N70 G77 N10                           Branch to sequence N10
N80 T02 D02 M06 (CUTTER DIAMETER=7)
N90 S1000 M40 M03
N100 G00 X95 Y10 Z3                   Point e, approach
N110 G01 Z0 F50                       Approach on Z
$0 MACHINING SUBROUTINE %40
N120 G77 H40 S3                       Branch to subroutine %40
N130 G77 N10                          Branch to sequence N10
N140 M02
```

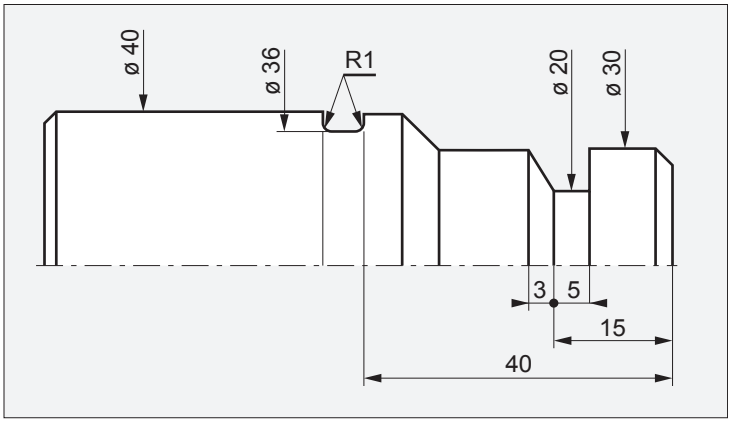
Subroutine for shape 1

```
%35
$0 SUBROUTINE OF MAIN PROGRAM %30
N10 G91 G01 Z-2.5 F40 M08          Point a, incremental infeed
N20 G90 Y40 F80                    Point b
N30 X50                            Point c
N40 G03 X50 Y10 I80 J25            Point d
N50 G01 X15                        Point a
```

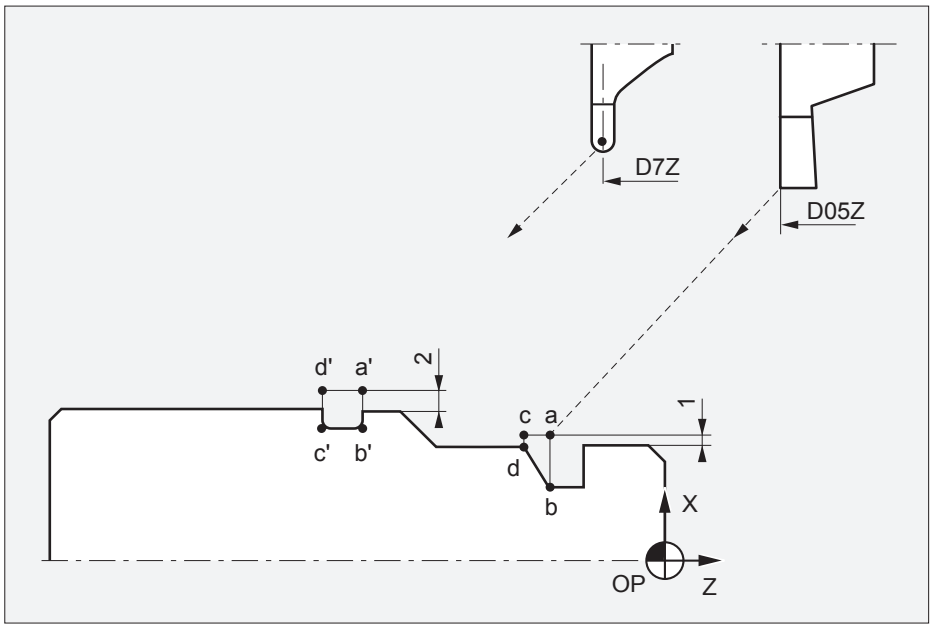
Subroutine for shape 2

```
%40
$0 SUBROUTINE OF MAIN PROGRAM %30
N10 G91 G01 Z-3 F40 M08            Point e, incremental infeed
N20 G90 Y80 F70                    Point f
N30 G02 X80 Y40 R15                Point g
N40 G01 X95                        Point h
N50 Y10                            Point e
```

Cutting grooves by calling two subroutines



Machining paths



Main program

```
%30
N10 G0 G52 X0 Z0
$0 EXTERNAL MACHINING
N..
N..
N110 G0 G40 G52 X.. Z.. G97 S900
$0 VEE-GROOVE
N120 T05 D05 M06 (TOOL WIDTH 5)
N130 X32 Z-15
N140 G96 S80
N150 G77 H35
N160 G77 N110
$0 GROOVE WITH RADIUS
N170 T07 D07 M06 (TOOL RADIUS 1)
```

Point a, approach

Call to subroutine %35
Call to sequence N110

```
N180 G0 G42 X44 Z-40          Point a', approach
N190 G96 S50
N200 G77 H40                  Call to subroutine %40
N210 G77 N110 M05             Call to sequence N110
N220 M02
```

Subroutine 1

```
%35
N10 G91 G01 X-6 G95 F0.08     Point b, incremental dimensions
N20 X6                        Outfeed to point a
N30 G0 Z-3                    Point c
N40 G01 X-1 F0.3              Point d
N50 X-5 Z3 F0.05              Point e
N60 X6 F0.08                  Point a
N70 G90
```

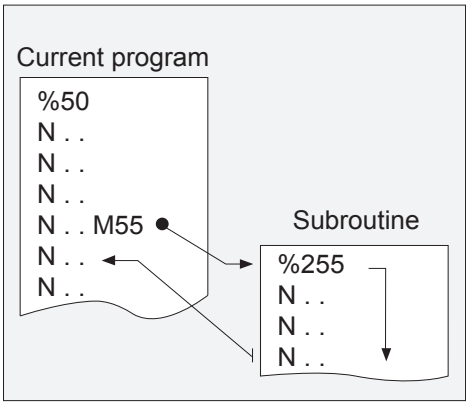
Subroutine 2

```
%40
N10 G91 G01 X-4 G95 F0.06     Point b', incremental dimensions
N20 Z-5 F0.08                 Point c'
N30 X4 F0.3                   Point d'
N40 G90
```

5.15.2 Subroutine Branch by M Function ► M

M..	Subroutine branch by M function.

The M.. function branches to a subroutine whose number is assigned by the machine manufacturer (see Machine operation manual).



Syntax

N.. **M..**

M.. Subroutine branch by M function.

Property of the Function

Function M.. programmed is modal however the subroutine is only executed once.

Notes

A subroutine called by an M function cannot itself call another subroutine by an M function. However, it can be nested with another type of call (by G function or PLC), but in no case may two calls of the same type be nested. Error 253 will be generated in case of nesting such subroutines.

A block including a subroutine call by M function may also include a positioning command.

Example: N.. M175 G00 X100

In this case, the block is processed in the following order:
- M175 sent to the PLC
- Movement X100 executed
- Call to the subroutine defined in machine parameter P35.

Return to current program

After execution of the subroutine, none of the data programmed earlier are restored.

The following data must be reprogrammed if necessary:
- modal preparatory G functions,
- modal technological S functions and miscellaneous M functions,
- correction D even if the tool was not changed,
- program variables L.

Context

The calling program context can be saved at the beginning of the called program. It is then restored at the end of execution of the called program. The programmed status access symbols are used for save and restore.

5.15.2.1 Specific Feature of a Call to a Subroutine Number Between %11001 to %11999

When the subroutine number is between %11001 and %11999, the subroutine is called before execution of the block containing the M function and block concatenation is forced. It should noted that within the subroutine, execution of the functions is enabled by G998 and/or G997 as in subroutine calls by Gxxx function.

Example:

Execution of spindle indexing during axis movement in a block containing the programming: N.. X.. Y.. Z.. M19.

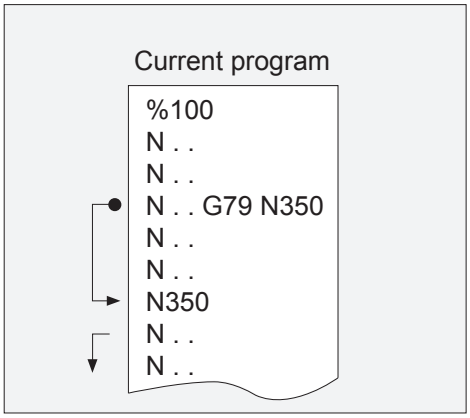
For further explanation on block concatenation and G997 / G998 please refer to Flexium Extended Programming manual M00020.

```
%11019 (SPINDLE INDEXING DURING MOVEMENT)
VAR     [br_pos] = [.BM62] *2+ [.BM65] *2+ [.BM64] - [.BM63] / 2&3+93524
        [M998] = [.BM999] - [.BM997] +998
        [tix(9)] [i]
ENDV
M997 (forced block sequencing)
FOR [i] = 1 TO 9 DO [tix(i)] = [.IBX(i)]
   BCLR [.IBX(i)] (stop movement)
ENDF
G997 (exit M19)
G999
FOR [i] = 1 TO 9 DO
   IF [tix(i)] = 1 THEN
        BSET [.IBX(i)] (enable movement)
   ENDI
ENDF
G997 (start movement)
(wait until spindle is in position)
WHILE E [br_pos] = 0 DO G4 F.1
ENDW
M[M998]
```

5.15.3 Branch to a Sequence without Return ► G79

G79	Conditional or unconditional branch to a sequence without return.

A conditional or unconditional branch is made to sequence number N.. associated with the function.



Syntax

N.. **G79** [L../E.. > = < Number] N..

G79	Conditional or unconditional branch to a sequence (an upstream or downstream branch is allowed).
L.. or E..	Variable L or parameter E tested in the condition (see chapter. 7.1 and 7.2).
> = <	Comparison symbols for the condition (up to two symbols can be used).
Number	Numerical expression of the condition.
N..	Mandatory argument defining the sequence number to which the jump is to be made.

Property of the Function

Function G79 is nonmodal.

Cancellation

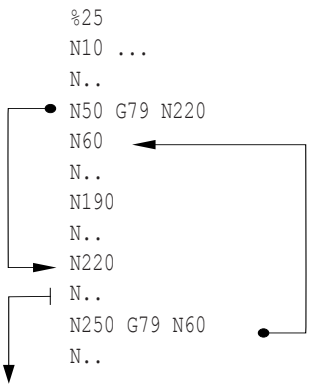
Function G79 is cancelled at the end of the block.

Notes

If the jump is conditional, the condition must be located between G79 and N.

Examples

Unconditional jump



```
%25
N10 ...
N..
N50 G79 N220
N60
N..
N190
N..
N220
N..
N250 G79 N60
N..
```

Jump to sequence N220

Jump to sequence N60

Conditional branch

Counting the number of machining operations performed and branching when the specified number is reached.

```
%75
N10 L5 = 20
N20 G52 Z..
N30 T01 D01 (TOOL)
N40 S800 M40 M03
N50 G00 X.. Y.. Z..
N60 G01 Z..
N70
N..
N..
N..
N140
N150 L5 = L5 -1
N160 G79 L5 = 0 N180
N170 G79 N50
N180 G52 Z.. M05
N190 M02
```

Variable L5 set to 20

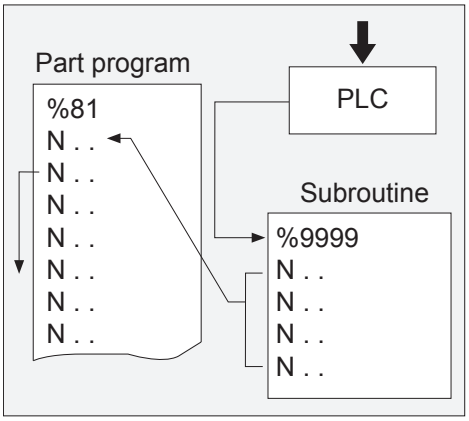
Programming for one machining operation

Count each operation down
Condition: If L5=0, branch to block N180
Branch to N50

5.15.4 Subroutine Call by PLC

During execution of a part program, the PLC can generate a call to a specific subroutine. The subroutine number is always %9999. If it exists, this functionality has been implemented by the machine builder.

For further information, please refer to the machine operation manual.



Condition for the branch to subroutine %9999

Part program being executed in mode:

- automatic (AUTO),
- single step (SINGLE),
- rapid (RAPID).

A subroutine called by the PLC cannot itself call another subroutine by PLC. However, it can be nested with another type of call (by M function or G function), but in no case may two calls of the same type be nested. If a part program is already being executed, the branch to subroutine %9999 is only made at the end of an interruptible block or the end of execution of an uninterruptible block.

Definition

An uninterruptible block is a block containing a G function causing a branch to a subroutine (G31, G45, G81, etc.) or a block for which certain parameters must be known in order to execute the next block (sequencing of three blocks in PGP mode). If no part program is being executed when subroutine %9999 is called, the CYCLE stops at the end of execution of the subroutine (the part program is not executed). Functions G01 and G40 are set at the start of the subroutine.

During execution of subroutine %9999

The PLC ignores a new call or maintaining of the call to subroutine %9999 during execution of this subroutine.

End of subroutine %9999

At the end of execution of the subroutine, the system does not report to the PLC. It is the subroutine's responsibility to report on its execution by function M or external parameter E.

Return to current program

After execution of the subroutine, none of the data programmed earlier are restored.

The following data must be reprogrammed if necessary:

- modal preparatory G functions,
- modal technological S functions and miscellaneous M functions,
- correction D even if the tool was not changed,
- program variables L.

Context

The calling program context can be saved at the beginning of the called program. It is then restored at the end of execution of the called program. The programmed status access symbols are used for save and restore.

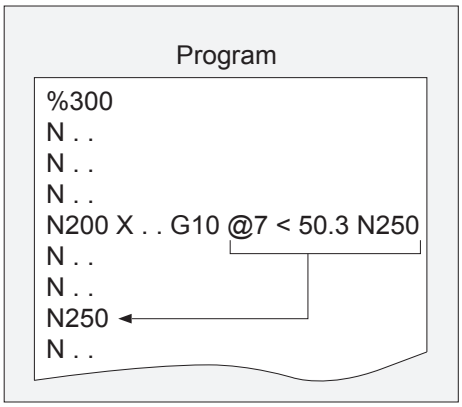
5.15.5 Block Interrupt ► G10

G10	Block interrupt.

G10 blocks are designed to be interrupted by a hardware input or by satisfying a position threshold for a particular axis.

After the interrupt, a branch may be arranged to a designated block. If no interrupt is detected, the next block is executed.

If an interrupt occurs, all axis positions are stored.



Note

The order of the data is important.

Syntax

N.. [G40] [G04 F..] [G00/G01/G02/G03] X.. Y.. Z.. **G10** [:n] [+X.. or F..]
[@n < > Value] N.. [+ Number] [EF..]

G40	Tool radius offset cancel.
G04 F..	Interruptible dwell, but without branch to designated block.
G00/G01/G02/G03	Interruptible interpolations.
X.. Y.. Z..	Interruptible axes.
G10	Block interrupt function.
:n	Numerical argument (number of interrupts from 1 to 99) significant only for hardware interrupts (on-the-fly dimension measurement). The interrupt block is not acknowledged (forcing the end of block) until the nth hardware interrupt.
+X.. or F..	Arguments defining a movement or dwell to be made after the interrupt(s). X..: A further movement to be made after the interrupt(s) before deciding whether to branch (possible on all the axes of the system, both measured and servoed). F..: Dwell in secs during which the block may be interrupted.
@n < > Value	Argument defining a condition for comparison with a threshold. @n: Physical address of the axis concerned by the test (axis address n is between 0 and 31). < >: Mandatory comparison symbol (Less than or greater than). Value: Comparison threshold in the same unit as the other axes of the channel (mm or inches).
N.. + Number	Branch to block N.. after the interrupt, possibly followed by a further number to reach unnumbered blocks.
EF..	Maximum feed rate after an interrupt.

Property of the Function

Function G10 is nonmodal.

Cancellation

Function G10 is cancelled at the end of the block.

Notes

All the arguments for function G10 are optional.

5.15.5.1 Analysis of the interrupt

When function G10 is programmed, an external interrupt may prematurely terminate the move. The end point programed will then not be reached. This signal stops the movement and replaces the dimensions of the required position by the dimensions of the current position.

If a sequence number (N address within the block data, not block number of the considered sequence) was programmed in the same block as the G10 function, then the system will jump to this sequence number if and only if an interrupt has been triggered.

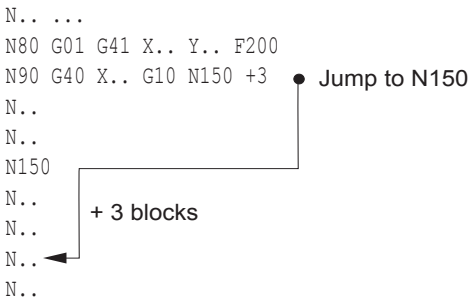
When interrupt IT is generated, the dimensions of all axes in the channel are stored in external parameters E7x001 (axis position reference where x = physical address of the axis). The processed interrupt is taken into account by function G10 of the part program that then manages the branch.

New dimensional values can be detected by dynamic operators and stored in an indexed addressing table.

Examples

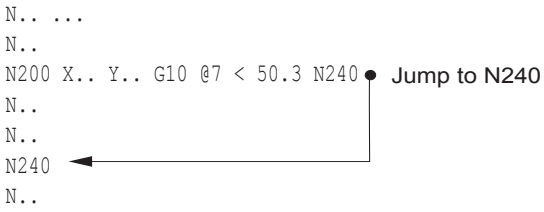
G10 block with a jump to an unnumbered block

Jump to the third block after N150 if a hardware interrupt is set.

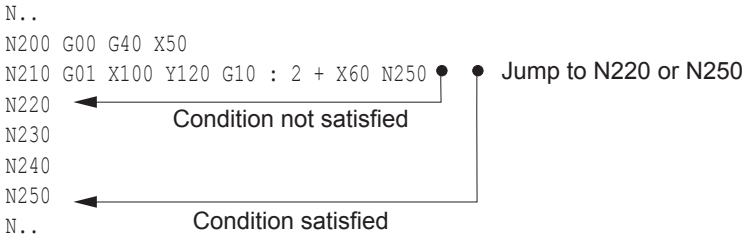


Comparison with a threshold and jump to a block once the condition is satisfied

Condition: Jump to block N240 when the measurement on axis 7 is less than 50.3.

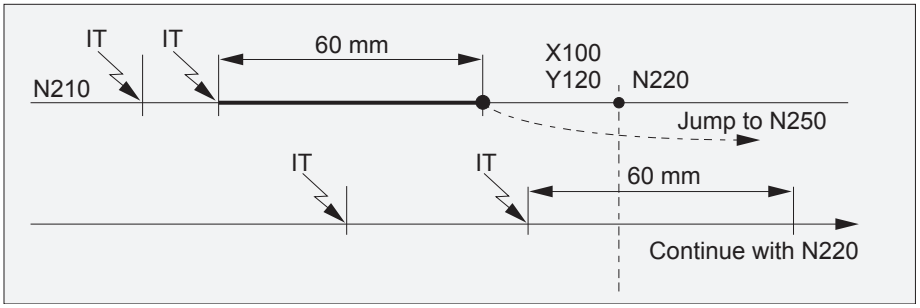


Expecting two interrupt + travelling an additional distance



If 60mm beyond the second interrupt the end point is beyond the original programmed target:
- the jump condition is not satisfied and the program continues with the next block.
- no updates are made to the E7x001 positions.

Diagram:



5.15.5.2 Special Use of Sequence Interrupt

Block Sequencing without Stopping Movement

In M999 status the analysis of the following blocks is suspended until the interrupt is detected.

When the interrupt is detected, if the current interpolated position (to which is added the +X value programmed after G10) is beyond the programmed end point, it should be noted that:

- analysis of the following blocks is resumed immediately,
- in case of possible processing by parametric programming, the current block is considered ended, thereby authorising read or write of external parameters E without effectively stopping current movements (this allows calculation of a positioning point according to the interrupt point and taking into account the geometric transformations made downstream of the interpolator),
- as soon as the next block is ready, the feed rate at the end of the current block is updated according to whether the path has changed or not.

In state M999, if the block was not interrupted, a subroutine call by the PLC is accepted at the end of execution of the block.

Feed Rate Limiting After an Interrupt

Before the interrupt, the speed of movement in block G10 is the modal feed rate F programmed before G10. After the interrupt, the feed rate at the end of the current block may be different if argument EF and its value (less than F) are programmed in that block or an earlier block.

Example

```
N..
N.. G01 X200 F300
N..
N.. G00 X..
N100 G01 X0 F150 G10 +X2 EF100        Feed rate 100 mm/min after the interrupt
N..
```

It is possible to measure the axes positions on an interrupt signal without having to program G10 and therefor without block interruption or branches. This is accomplished through parameter E11019 and is explained more in details in the extended programming manual.

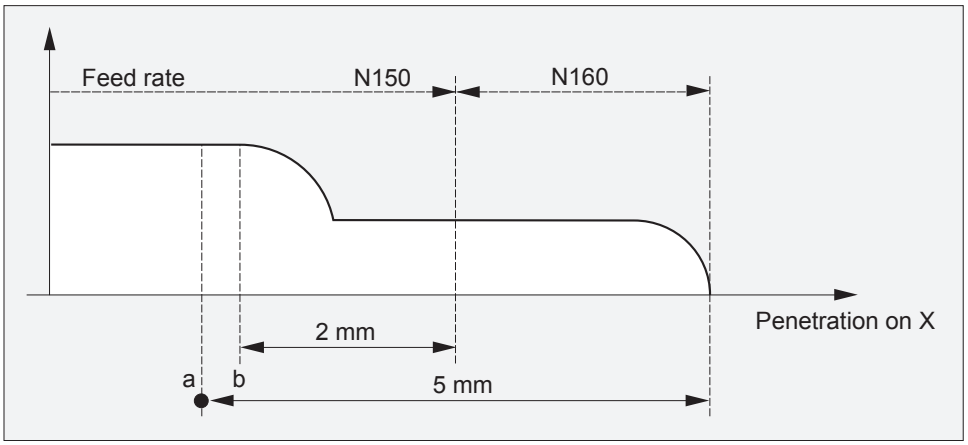
Example

The example below is defined for a system using inclined axis programming.

For the specific data used in the example, refer to the additional information given in this manual for:
- Inclined axes
- External parameters E...
- Symbolic variables [...].

```
%398
N..
N..
(Compute the offset between the machine dimension and program dimension in the program reference
system)
[K] = CE69001 [DECX] = UE70000 * [K] - [.IRX(1)]
N150 G01 X0 F200 G10 +X2 EF100
(Go to 5 mm from the interrupt point)
N160 [COTE] = [K] * UE70001 - [DECX] - 5 X[COTE] F100
N..
```

In the figure:
- point a: point measured on the interrupt
- point b: current position interpolated at the time of the interrupt.



5.15.6 Temporary Suspension of Next Block Preparation ► G79+/-

G79 +/-	Temporary suspension of next block preparation in a sequence with movements.

Programming this function in a block with movements temporarily suspends preparation of the next block. A lead or lag (expressed in millimetres or seconds) at the beginning of the current block or at the end of execution of the current block is programmed on the restart of block processing.

Syntax (XY plane)

N.. [G00/G01/G02/G03] X.. Y.. Z.. **G79 +/-** X.. / F..

G00/G01/G02/G03	Linear or circular interpolation.
X.. Y.. Z..	End point.
G79	Temporary suspension of next block preparation in sequences with movement.
+/-	+: Lag with respect to the start of the current block.
	-: Lead with respect to the end of execution of the current block.
X.. / F..	X..: Lag or lead in millimetres.
	F..: Lag or lead in seconds.

Properties of the function

Function G79 is nonmodal.

Cancellation

Function G79 is cancelled at the end of the block.

Restart of processing after programming of the function

As soon as analysis of the next block is restarted, it should be noted that:

- in case of possible processing by parametric programming, the current block is considered ended, thereby authorising read or write of external parameters E without effectively stopping movements,
- as soon as the following block is ready, the feed rate at the end of the current block is updated,
- this allows write of external parameters E and discrete outputs or read and test of these parameters without stopping movements.

Examples

Timeout at the start of the current block

```
N..
N110 G00 X0 Y0
N120 G01 X100 G79+F4
N130 G79 E10000=1 N300
N140 Y20
N..
N300 G01 X200
N..
```

Movement along X with four second delay before analysis of block N130

Anticipation at the end of the current block

```
N..
N110 G00 X0 Y0
N120 G01 Y100 G79-X10
N130 ...
```

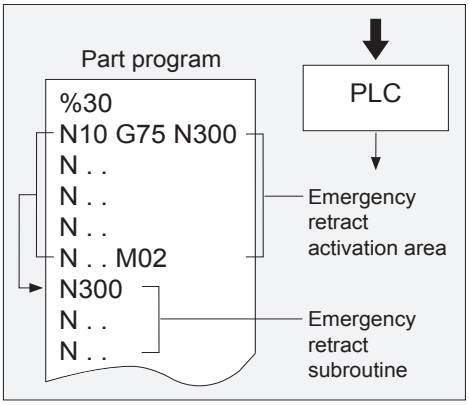
Movement along Y with anticipation of 10 mm before analysis of block N130

5.15.7 Emergency Retract ► G75

G75	Declaration of an emergency retract subroutine.

The emergency retraction subroutine is activated by a request from the PLC.

Activation stops the current program and causes a branch to address N.. of the last emergency retract subroutine declared.



Syntax

N.. **G75** N..

G75	Declaration of an emergency retract subroutine.
N..	Mandatory argument associated with the function and giving the address of the emergency retract subroutine start sequence number.

Properties of the Function

Function G75 is nonmodal. Argument N.. associated with the function is modal.

Cancellation

Declaration of a subroutine G75 N.. is cancelled by:
- cancellation function G75 N0,
- function G75 N.. with a different subroutine number,
- end of program function M02,
- a reset.

Condition for activation of the emergency retract subroutine

Part program being executed in one of the following modes:
- automatic (AUTO),
- single step (SINGLE),
- manual data input (MDI),
- rapid (RAPID).

In automatic mode, the emergency retraction program remains active until encountering a function M00 or M02.

An emergency retraction program can be activated:
- by reading of the block in which it is programmed,
- as long as the program or subroutine in which it is declared is not ended,
- as long as a new emergency retraction program has not been declared,
- as long as the emergency retraction function has not been cancelled by G75 N0.

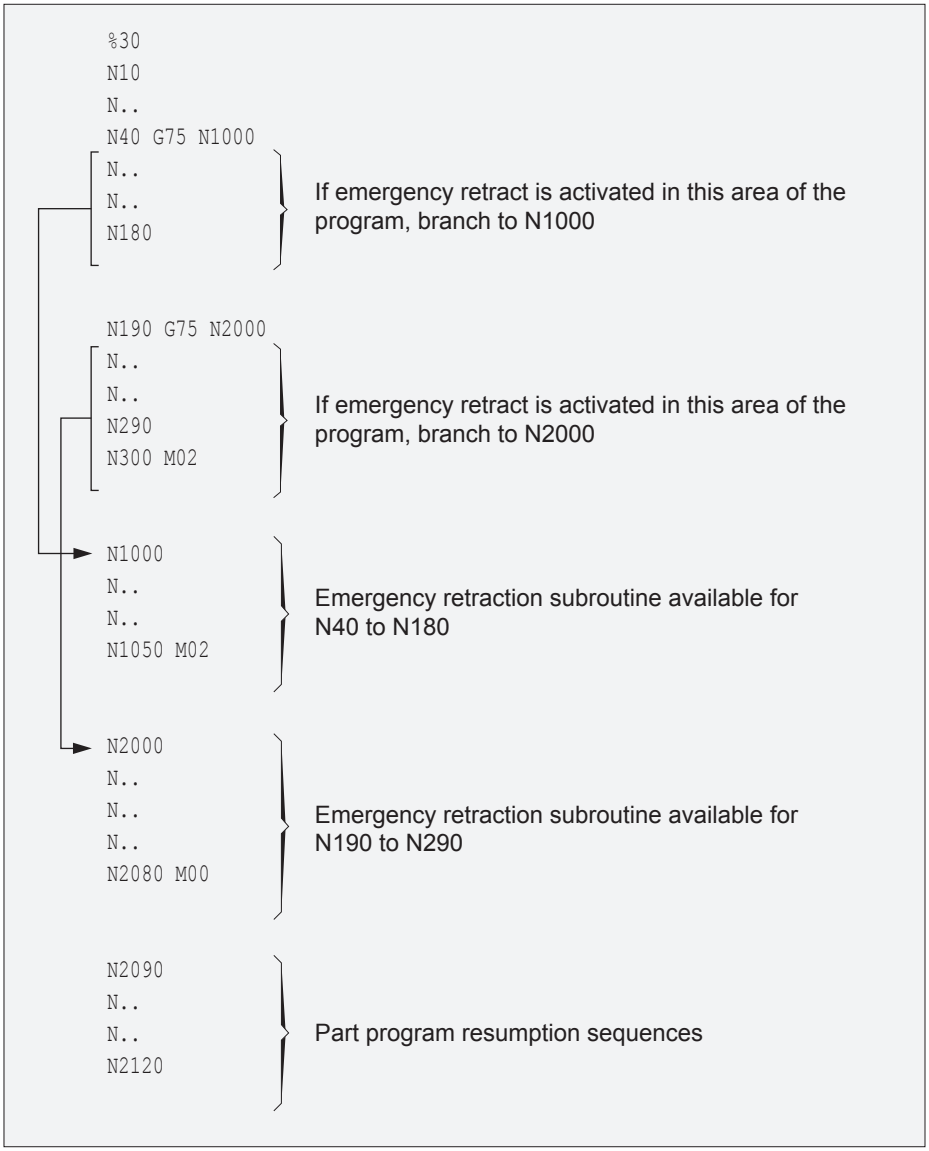
If emergency retraction is activated but no retraction program has been declared in the part program, it has the same effect as pressing Cycle stop key.

Activation of an emergency retraction is sent by the PLC bit:
Flexium_NCK.W_ALLCNC.EmergencyRetract.

It can then be generated automatically or on an operator request (see machine operation manual).

Example

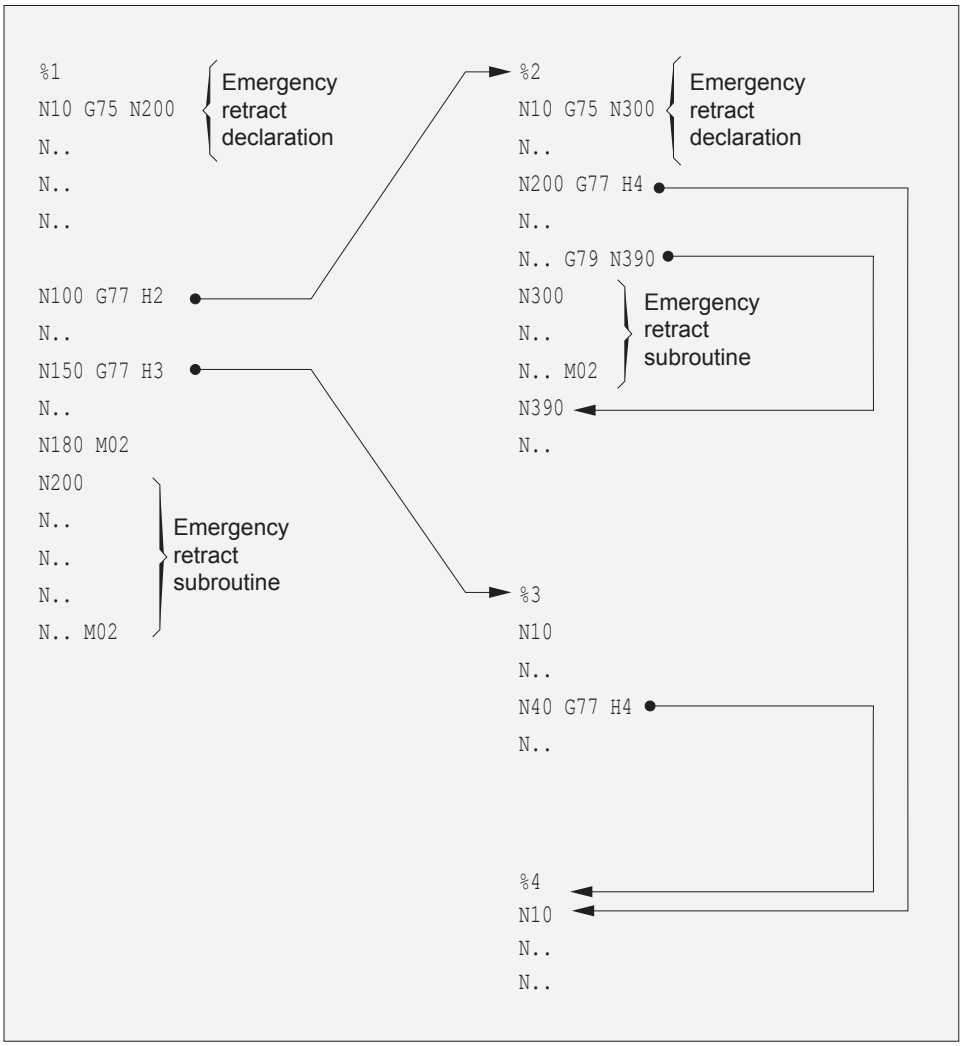
Declaration of emergency retract subroutines from a main program



Declaration of emergency retract subroutines from a program including subroutine branches

If an emergency retract subroutine is activated during execution of subroutine %4 called by N200 of %2, branch to sequence N300 of %2.

If an emergency retract subroutine is activated during execution of subroutine %4 called by N40 of %3, branch to sequence N200 of %1.



5.15.8 Branch to Automatic Homing Subroutine %9990

Homing can be done automatically on each of the machine axes by running subroutine %9990.

Conditions for running subroutine %9990

Subroutine %9990 is run by pressing the cycle start button:

- if the system is in homing mode,
- if no other program is being executed.

During execution of subroutine %9990

As soon as subroutine %9990 or one of its subroutines starts, the machine axes are authorised for programmed motion despite not being homed.

End of execution of subroutine %9990

At the end of execution of subroutine %9990 (by programming of M02), the program that was active when %9990 was called is re-instated.

5.15.9 Subroutine Branch on a Reset %11000

On a reset, the NC channels can execute subroutine %11000.

Conditions for running subroutine %11000

Subroutine %11000 is run:
- if activated by the machine builder,
- if this subroutine is present in the memory.

Start of execution of subroutine %11000

Subroutine %11000 begins approximately 100 microseconds after the reset pulse (S RAZ or E_RAZ) goes high. It should be noted that this pulse remains high until the subroutine ends.

Contents of subroutine %11000

A subroutine %11000 must not include any executable functions such as:
- Movements on the axes
- Dwells (G04 F..)
- Miscellaneous functions (M..), etc.

Only the following operations are allowed in subroutine %11000:

- read and write of external parameters E and program variables L (see chapter 7). It should be noted that at the end of execution of the subroutines, variables L0 to L19 are reset but the others preserve their values,
- declaration of dynamic operators and enabling of interaxis calibration (see Flexium Extended Manual M00020),
- declaration and initialisation of symbolic variables [...]. It should be noted that they must be saved by the SAVE function to be available after %11000 has ended.

5.15.10 Restrictions Related to Drip Feed Mode

Drip feed mode allows a program to be checked or to be executed while being simultaneously transmitted via the machine panel or the network.

Drip feed mode is used for execution of large programs which cannot be stored in full in the NC.

Restrictions related to drip feed mode

It is impossible to use functions referring to block numbers in the program:

- jump to a sequence (G79 N..),
- block interrupt (G10 N..),
- branch to sequences of the main program (G77 N.. N..),
- activation of an emergency retract subroutine (G75 N..).

Branches to subroutines stored in the memory are allowed in drip feed mode (G77 H.. or G77 H.. N.. N..).

5.15.11 Call to Subroutine Return Block ► G77 -i

G77 -i	Call to subroutine return block.

This function allows a subroutine to call and execute instructions held in the return block of the calling subroutine.

Syntax

N.. **G77 -** i

G77	Call to subroutine return block.
-i	Immediate value (or variable) giving the nesting level of the program containing the block called with respect to the nesting level in which G77 -i is programmed (the return block called in this way is automatically postincremented).

Properties of the Function

Function G77 -i is nonmodal.

Cancellation

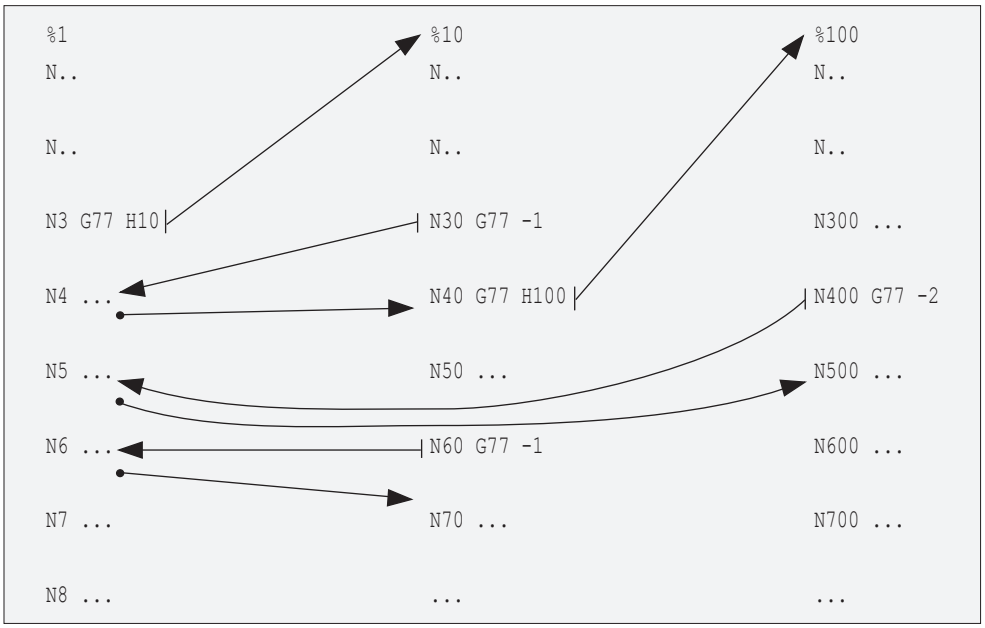
Function G77 -i is cancelled at the end of the block.

Notes

If the block called is in the «displayable» program, it is displayed on the program page (PROG) even if the subroutine that called it is not displayable.

It should be noted that:
- if the nesting level requested is not accessible, the system returns error message 26
- the blocks called by G77 -i must not include branches or other calls (G79, G77, etc.) or the system returns error message 26.

Example



5.15.12 ISO Program or Block Creation/Deletion ► G76+/-

G76+/- ISO program or block creation/deletion.

General

Programming of function G76+/- provides the following capabilities:
- Program creation,
- Program deletion,
- Block insertion in a program,
- Block deletion from a program.

This function is especially useful for creating part programs automatically (e.g. after teach-in by gauging of dimensions).

This function can only be used to create and delete programs and blocks located in part program area zero.

Before execution of a function G76+ (creation, insertion), the system makes sure no other program is active. If so, program execution is suspended as long as the other program remains active.

With G76, the program number can be indexed (e.g. H123.2). This is accepted providing no other program in any of the program areas has the same number.

If there is not enough memory space for program creation or block insertion, the system returns error message 266.

If the block including function G76 has a format error, the system returns error message 1.

Properties of the Function

Function G76 is nonmodal and is cancelled at the end of the block.

Program created or edited via G76 are stored in dynamic RAM.
They can be saved either by transfer to the panel or by a proper shutdown of the CNC system. If the system is powered off abruptly and the file has not been transferred the content is lost.

5.15.13 Creating a Program ► G76+ H

The syntax below is used to create a program in area zero.

Syntax

N.. **G76+** H..

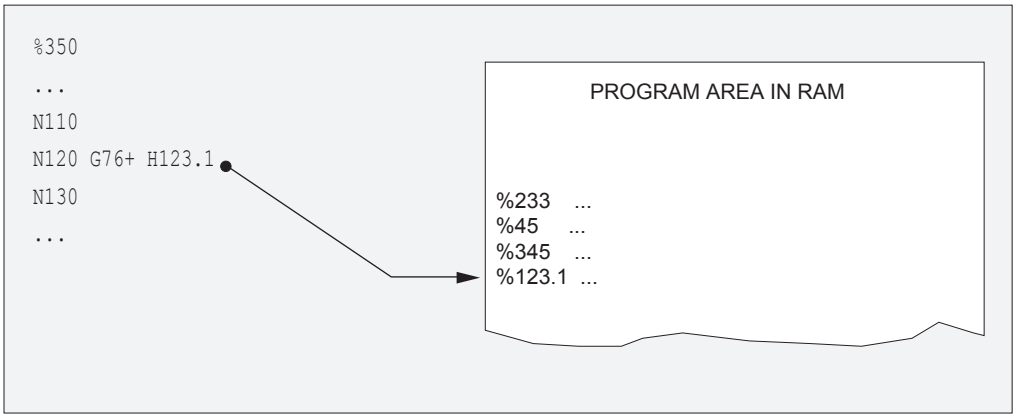
G76+	Program creation function.
H..	Number of the program to be created.

Notes

In program creation mode, the program number must be the last word in the block.

Example

Program creation in User memory zone.



5.15.14 Deleting a Program ► G76- H

The syntax below is used to delete a program located in area zero.

Syntax

N.. **G76-** H..

G76-	Program deletion function.
H..	Number of the program to be deleted.

Notes

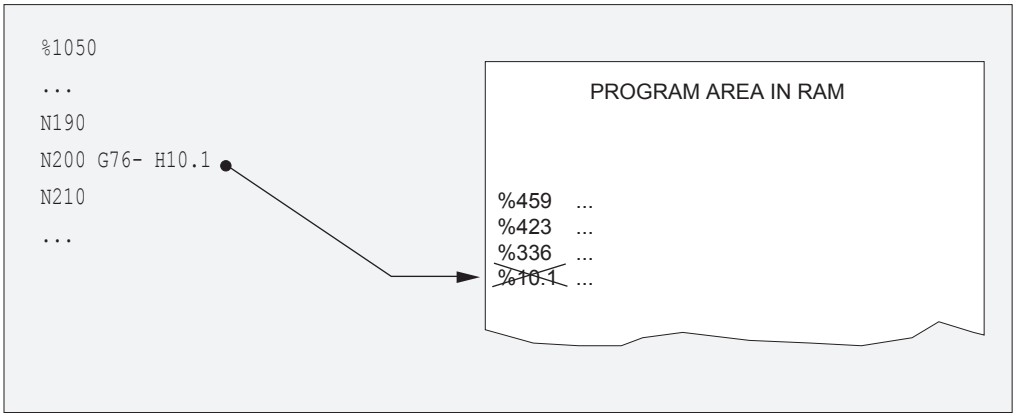
The program number must be the last word in the block.

Several cases can occur, depending on the area where the program is located:

- If a program in area zero has the number specified in the function, this program is deleted,
- If a program in an area other than area zero has the number specified in the function, deletion is refused and the system returns error message 266,
- If no program in any area has the number specified with the function, the command is accepted (but no program is deleted).

Example

Program deletion in RAM program area (area zero).



5.15.15 Inserting a Block ► G76+

The syntax below is used to insert a block in an existing program.

Syntax

N.. **G76+** [H..] N..[+number] ISO block

G76+	Function defining block insertion.
H..	Specifies the program number where the block is to be inserted (optional: default H..: the ISO block is inserted in the program including function G76).
N.. +number	N..: Block number (mandatory). The insertion is made after this block unless "+number" is specified (note: the first block of program %.. is block N0). +number: (optional) defines the number of lines (after the block number specified) where the insertion is to be made.
ISO block	Block to be inserted consisting of ISO functions (see list under Notes).

Notes

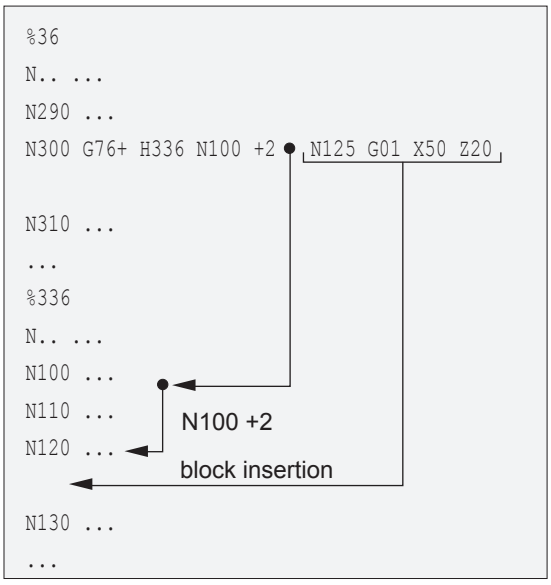
The following ISO functions are accepted:
- N.. (block number),
- G.. (preparatory functions; several G functions are accepted in the block),
- X.. Y.. Z.. U.. V.. W.. A.. B.. C.. (axes and dimensions),
- I.. J.. K.. (circle centre coordinates),
- P.. Q.. R.. (material vector for 3D radius correction),
- /.. (polynomial coefficients),
- L=.. to L9=.. (program variables).

The values associated with the above functions are immediate integer or fractional values, program variables (L..), symbolic variables [sym] or parameters E. When they are parameters E, the values are specified in the internal system unit (these data are edited in the program as immediate values, signed if negative, and in the internal unit defined for linear or rotary axes).

The block to be inserted must not have more than 120 characters or the system returns error message 1. If there is not enough memory space for block insertion, the system returns error message 266. If the block to be inserted does not exist, the system returns error message 25.

Example

Block insertion in program %336 located in the RAM program area (area zero).



5.15.16 Deleting a Block ► G76-

Syntax

N.. **G76-** [H..] N..[+number]

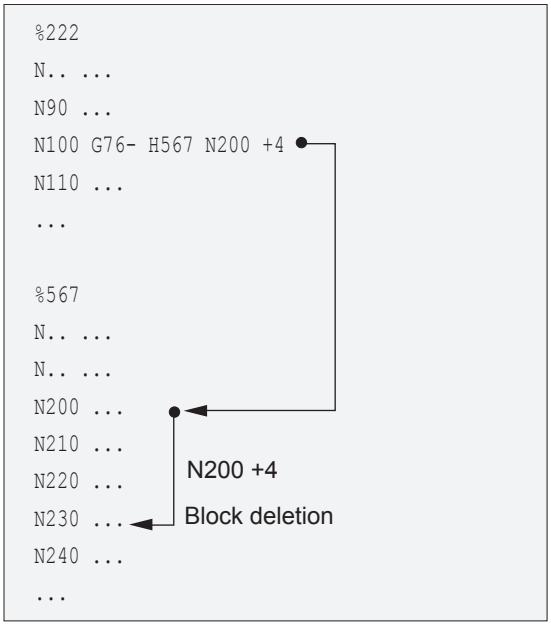
G76-	Block deletion function.
H..	Specifies the program number from which the block is to be deleted (optional: default H..: the ISO block is deleted from the program including function G76).
N.. +number	N..: Block number (mandatory). It is this block that is deleted unless +number is programmed (note: the first block of program %.. is block N0). +number: (optional) specifies the line to be deleted (after the block number specified).

Notes

If the block to be deleted does not exist, the system returns error message 25.

Example

Deletion of a block in program %567 located in the RAM program area (area zero).



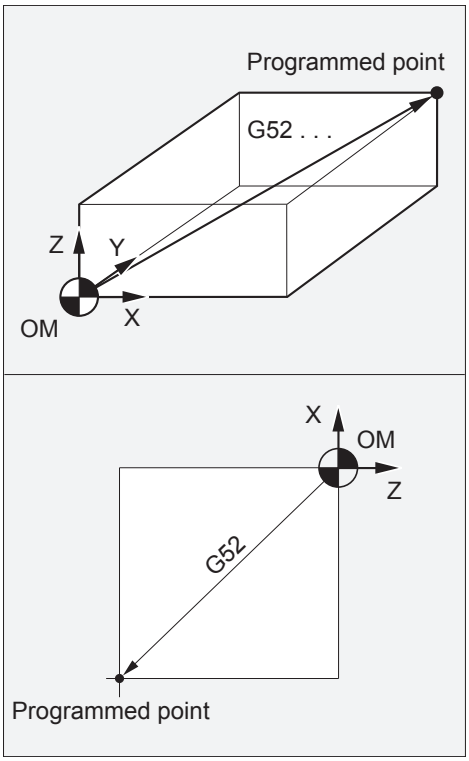
5.16 Movement Referential selection

5.16.1 Programming of Movements in Absolute Coordinates Referenced to the Measurement Origin ► G52

G52	Programming of movements in absolute coordinates referenced to the measurement origin.

Movements programmed with this function are referenced to the measurement origin (OM).

All the axes are programmable with reference to the measurement origin.



Syntax

N.. [G40] [G90] [G00/G01] **G52** X.. Y.. Z.. A.. B.. C.. [F..]

G40	Radius offset cancel.
G90	Programming in absolute dimensions.
G00/G01	Linear interpolations at high or programmed speed.
G52	Programming of movements in absolute coordinates referenced to the measurement origin.
X.. Y.. Z.. A.. B.. C..	End point referenced to the measurement origin.
F..	Feed rate.

Property of the Function

Function G52 is nonmodal.

Cancellation

Function G52 is cancelled at the end of the block.

Notes

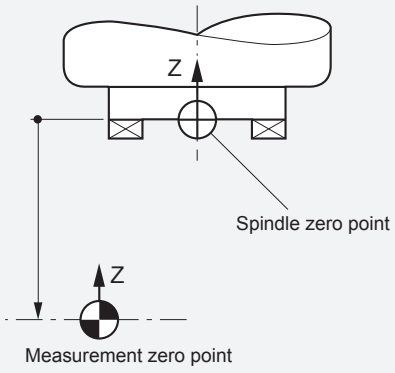
Programming function G52 in a block suspends the following data:
- tool data,
- DAT1,
- DAT2,
- DAT3,
- program origin offset (G59),
- angular offset (ED..),
- scaling factor (G74).

Function G52 must:
- precede programming of the axes in the block,
- be programmed with the system in G40 mode (radius offset cancel) or the system returns error message 27,
- be programmed in absolute coordinates (G90).

Examples

Programming the Z axis with reference to the measurement zero point before a tool change

```
%10
N10 G00 G52 Z0
N20 T03 D03 M06
N..
```



Programming carrier/carried axes pairs

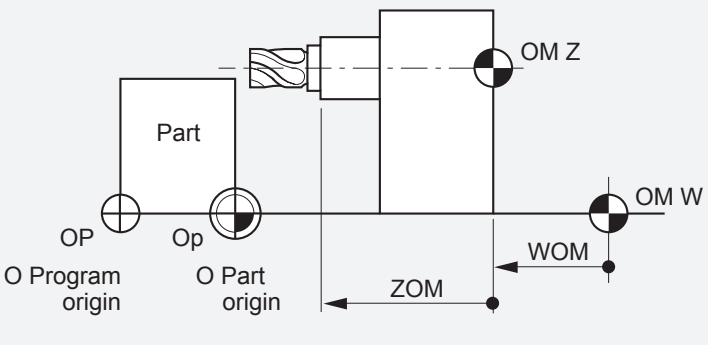
A parallel carrier/carried axis pair can be programmed in G00 and G52.

In all other cases programming a pair of carrier/carried axes simultaneously is prohibited.

Example

DAT1 and DAT2 do not apply to the values programmed with the Z and W axes.

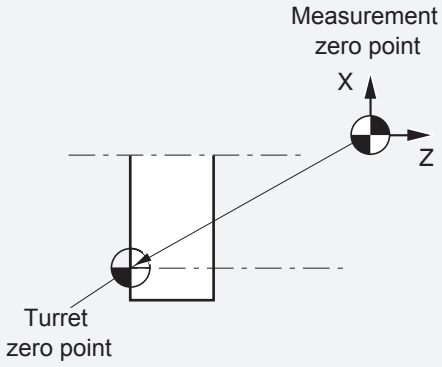
```
N..
N.. G00 G52 Z.. W..
N..
```



Examples

Programming X and Z axes with reference to the measurement zero point before a tool change

```
%10
N10 G00 G52 X0 Z0
N20 T03 D03 M06
N..
```



Programming of the carrier/carried axes

A parallel carrier/carried axis pair can be programmed in G00 with reference to the measurement origin. In all other cases, programming in G00 is prohibited.

Example

DAT1 and DAT2 do not apply to the values programmed with the Z and W axes.

```
N..
N.. G00 G52 Z.. W..
N..
```

5.16.2 Datum Shift DAT1 and DAT2 Cancel/Enable ► G53, G54

G53	DAT1 and DAT2 cancel.

G54	DAT1 and DAT2 enable.

These functions are used to enable or cancel DAT1 and DAT2 shifts entered on the context “Work Offset”.

Syntax

N.. **G53/G54**

G53	DAT1 and DAT2 cancel.
G54	DAT1 and DAT2 enable.

Properties of the Functions

Functions G53 and G54 are modal.

Function G54 is the default function.

Cancellation

Functions G53 and G54 cancel one another.

Notes

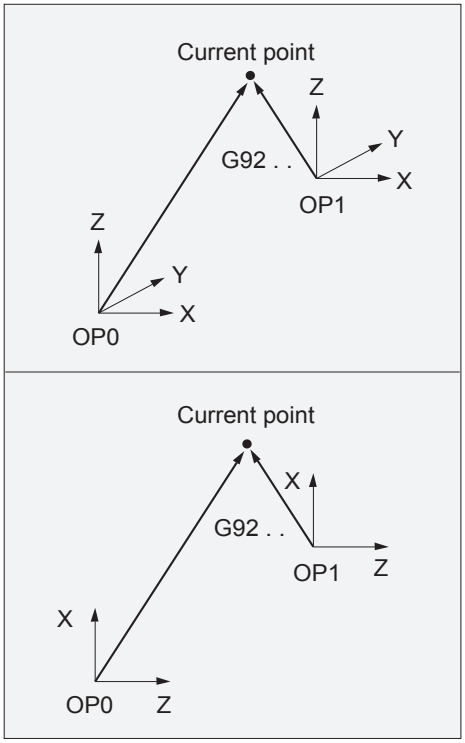
The tool offsets are not affected by function G53.

5.16.3 Program Origin Preset ► G92

G92	Program origin preset.

This function, associated with one or more axes and their values, defines the current point with reference to a new program origin.

DAT1 values are redefined for the programmed axes.



Syntax

N.. **G92** X.. Y.. Z..

G92	Program origin preset.
X.. Y.. Z..	Current position with respect to the new program origin.

Calculation for program origin preset on an axis

New DAT1 = Previous DAT1 + Previous current point/OP - Value programmed with G92

or

New DAT1 = Current point/OM - Value programmed with G92 - Tool length (in the axis) - DAT2

This operation is not performed until completion of the block before that containing function G92.

STOP The new offsets remain active at the end of the program.

Program origin preset function G92:
- applies to all the axes, both carried and independent,
- is rejected if the last movement was programmed with reference to the measurement origin: (error message 2),
- is ignored in test (TEST) and sequence number search (SEARCH) modes,
- suspends analysis of blocks until execution of the previous block is completed,
- cannot be programmed with radius offset; the system must be in state G40,
- cannot be programmed in PGP (Profile Geometry Programming).

Example

- Value entered for Z DAT1 = -300
- Value entered for Z DAT 2 = 20
- Tool length L (correction D9) = 80
- Program origin preset G92 Z60

```
N..
N150 G00 D9 G40 X0 Y0 Z40
N160 G92 Z60                    Preset
N170 G00 Z..
N..
```

Analysis of block N150 yields:

- current point/OM Z = -160

Application of the first equation

New Z DAT1 = -300 + 40 - 60 = -320

Application of the second equation

New Z DAT1 = -160 -60 -80 -20 = -320

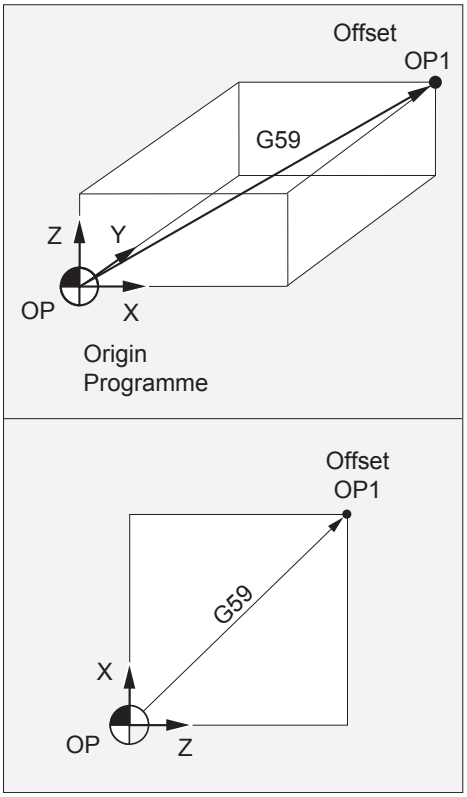
5.16.4 Program Origin Offset ► G59

G59	Program origin offset.

This function, associated with one or more arguments, axes and values, offsets the program origin (OP).

An origin offset can be assigned to each system axis.

No movement is caused by the function and its arguments.



Syntax

N..[G90/G91] **G59** X.. Y.. Z.. U.. V.. W.. A.. B.. C..[I.. J.. K.. ED..]

G90/G91	Programming in absolute or incremental dimensions.
G59	Program origin offset.
X.. Y.. Z.. U.. V.. W.. A.. B.. C..	The axes programmed are the arguments associated with the function. They must immediately follow the function and at least one axis must be programmed.
I.. J.. K.. D..	I.. J.. K..: Arguments defining the centre of rotation of an anglar shift programmed with ED in the plane with respect to the initial program origin). The translation, if any, of the program origin is carried out after the rotation. ED..: Angular shift.

Properties of the Function

Function G59 is nonmodal. The arguments associated with the function are modal.

Cancellation

A programmed offset G59.. is cancelled by:
- programming G59 with axis arguments assigned zero values in absolute dimensions (G90),
- the end of program function (M02),
- a reset.

Notes

For simplification and understanding of the program, it is recommended to be in state G90 (absolute programming) before programming an origin offset.

5.16.4.1 Function G59 programmed in absolute dimensions (G90)

The origin offset G59 ... is referenced to DAT1 + DAT2. A new origin offset G59 ... replaces a previous one.

5.16.4.2 Function G59 programmed in incremental dimensions (G91)

The first movement programmed after G59 ... is translated by the value of the program origin offset. A new origin offset is applied to the next movement, but the absolute position is offset by the sum of all the G59 ... functions programmed earlier.

Remark

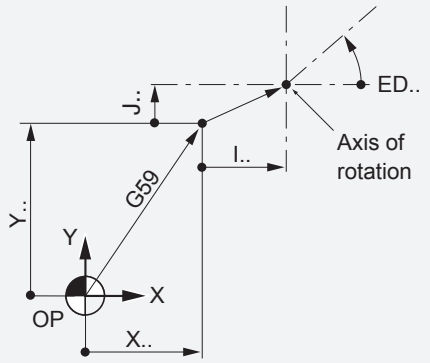
Regardless absolute or incremental programming, it is important to respect the following order of declaration:

- ED.. Angular offset,
- G59 ... Program origin offset,
- G51 ... Mirror function,
- Scaling factor.

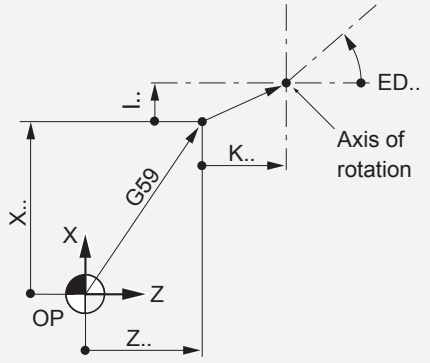
5.16.4.3 Features related to arguments I, J (K)

Machining programmed with respect to the program origin (OP) can be translated and oriented according to the angle programmed with ED (see chapter. 5.15.5). An offset on the X and Y (Z) axes is not mandatory for programming I and J (K).

```
N.. ...
N.. G59 X.. Y.. I.. J.. ED..
N.. ED..
N..
```

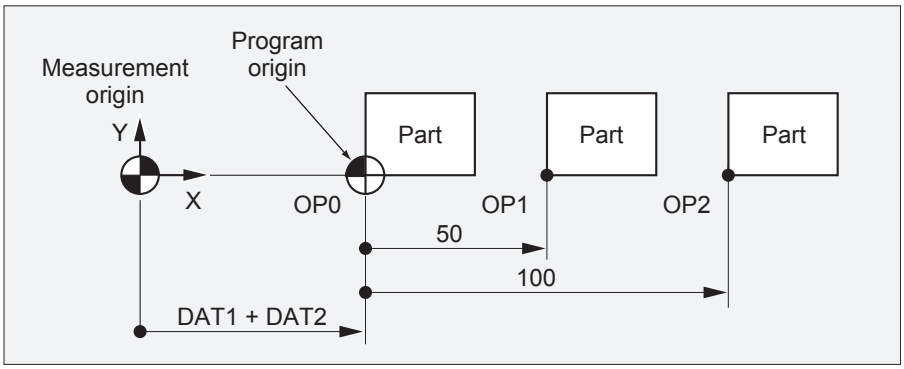


```
N.. ...
N.. G59 X.. Z.. I.. K.. ED..
N.. ED..
N..
```



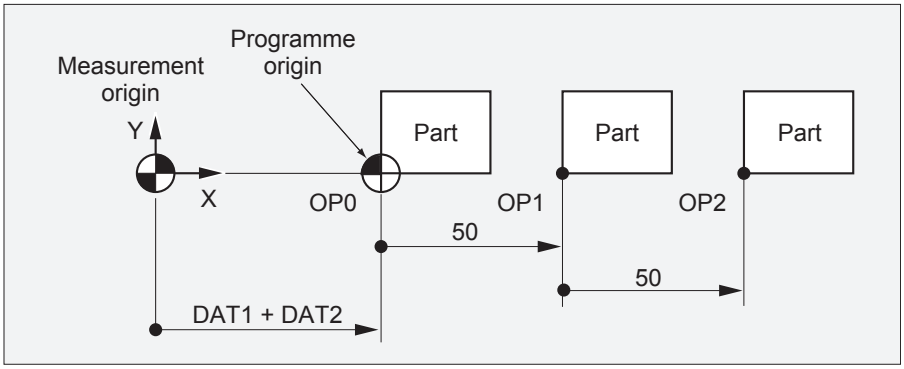
Examples

Origin offsets on the X axis programmed in absolute dimensions



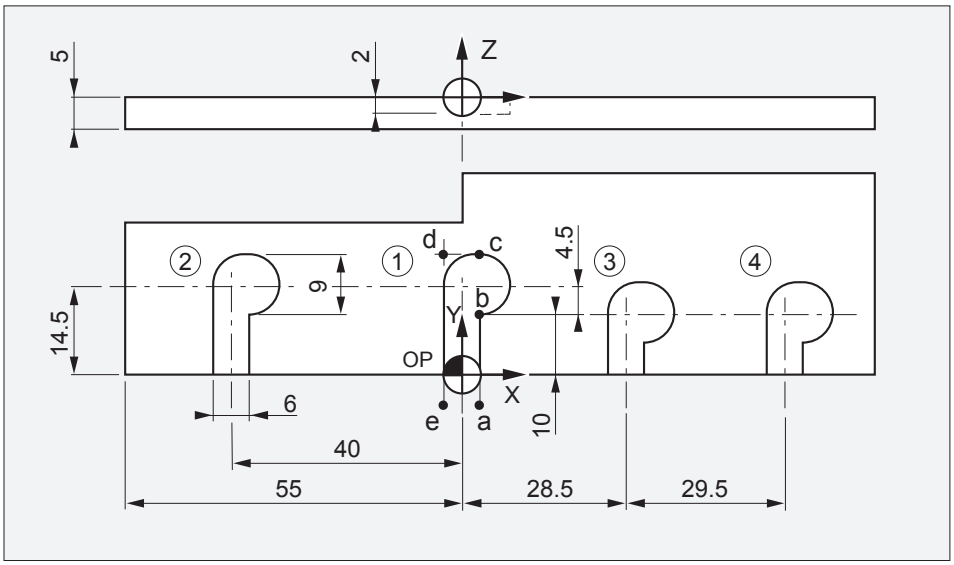

```
%60
N10
N..
N50
N..
N90
N..
N120 G90 G59 X50                    Offset 1
N.. G77 N50 N90                     Machining
N..
N230 G59 X100                       Offset 2
N.. G77 N50 N90                     Machining
N..
N350 G59 X0                         Cancellation
N..
```

Origin offsets on the X axis programmed in incremental dimensions



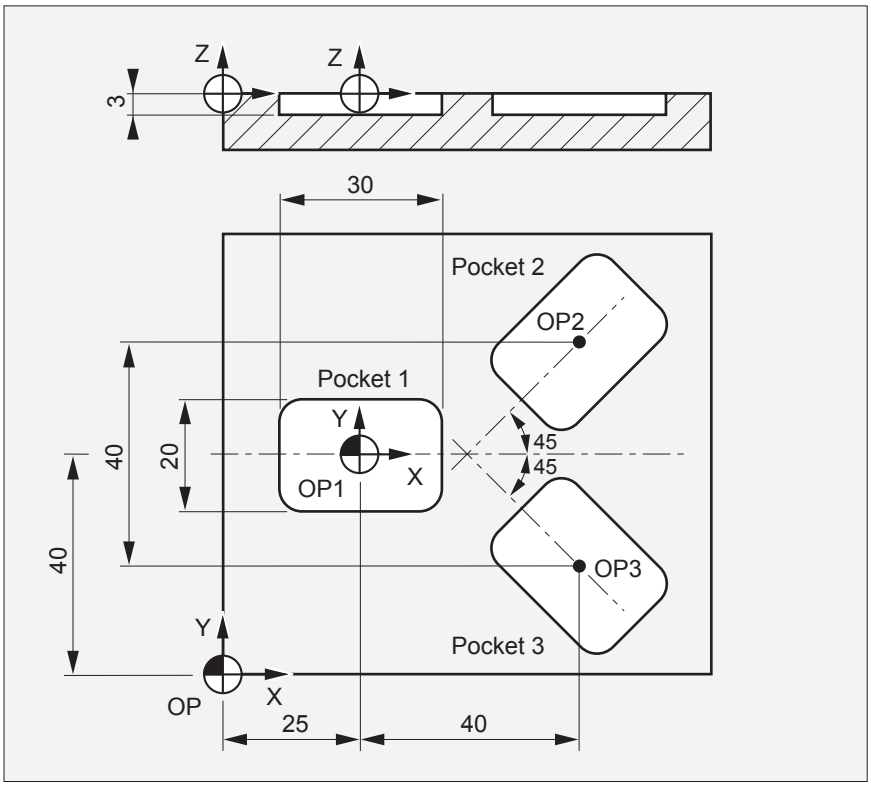

```
%70
N10
N..
N50
N..
N90
N..
N120 G91 G59 X50                    Offset 1
N.. G77 N50 N90                     Machining
N..
N230 G59 X50                        Offset 2
N.. G77 N50 N90                     Machining
N..
N350 G91 G59 X-100 or G90 G59 X0    Cancellation possible in G91 or G90
N..
```

Repetition of a shape with origin offsets, XY plane (G17) example




```
%110
N10 G00 G52 Z0
N20 T09 D09 M06 (CUTTER DIAMETER=5)
N30 S2000 M40 M03
$0 SHAPE 1
N40 G00 G41 X3 Y-5                  Point a, approach X, Y
N50 Z-2                             Approach on Z
N60 G01 Y10 F120                    Point b
N70 G03 X3 Y19 R4.5 F80             Point c
N80 G01 X-3 F120                    Point d
N90 Y-5                             Point e
$0 SHAPE 2
N100 G59 X-40                       Absolute offset on X
N110 G77 N40 N90
$0 SHAPE 3
N120 G59 X28.5 Y-4.5                Absolute offset on X and Y
N130 G77 N40 N90
$0 SHAPE 4
N140 G91 G59 X29.5                  Incremental offset on X
N150 G90 G77 N40 N90
N160 G59 X0 Y0                      Offset cancelled
N170 G00 G40 G52 Z0 M05 M09
N180 M02
```

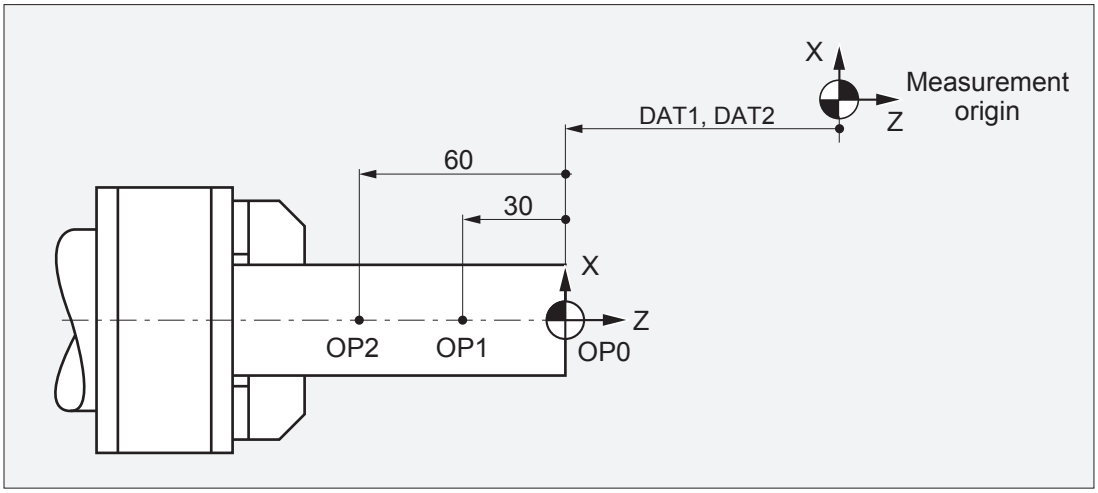
Repetition of a pocket by origin offsets and angular offsets




```
%61
N10 G00 G52 Z0
N20 T11 D11 M06 (CUTTER DIAMETER=10)
N30 S800 M40 M03 M08
$0 POCKET 1
N40 G59 X25 Y40                          Origin offset
N50 G45 X0 Y0 EX30 EY20 ER1 P3 Q8 Z-3
EP50 EQ200
$0 POCKET 2
N60 G59 X65 Y60                          Pocket centre offset
N70 ED45 G77 N50 N50                     Angular offset and machining
$0 POCKET 3
N80 G59 X65 Y20                          Pocket centre offset
N90 ED-45 G77 N50 N50                    Angular offset and machining
N100 ED0 G00 G52 Z0 M05 M09
N110 M02
```

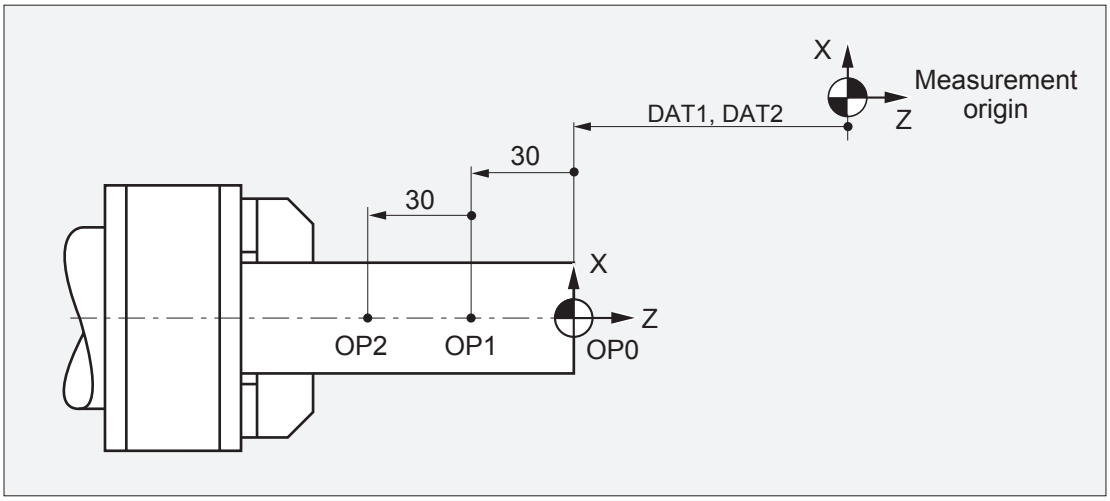
Examples

Origin offsets on the Z axis in absolute programming




```
%60
N10
N..
N50
N..
N90
N..
N120 G90 G59 Z-30          Offset 1
N.. G77 N50 N90            Machining
N..
N230 G59 Z-60              Offset 2
N.. G77 N50 N90            Machining
N..
N350 G59 X0                Offset cancelled
N..
```

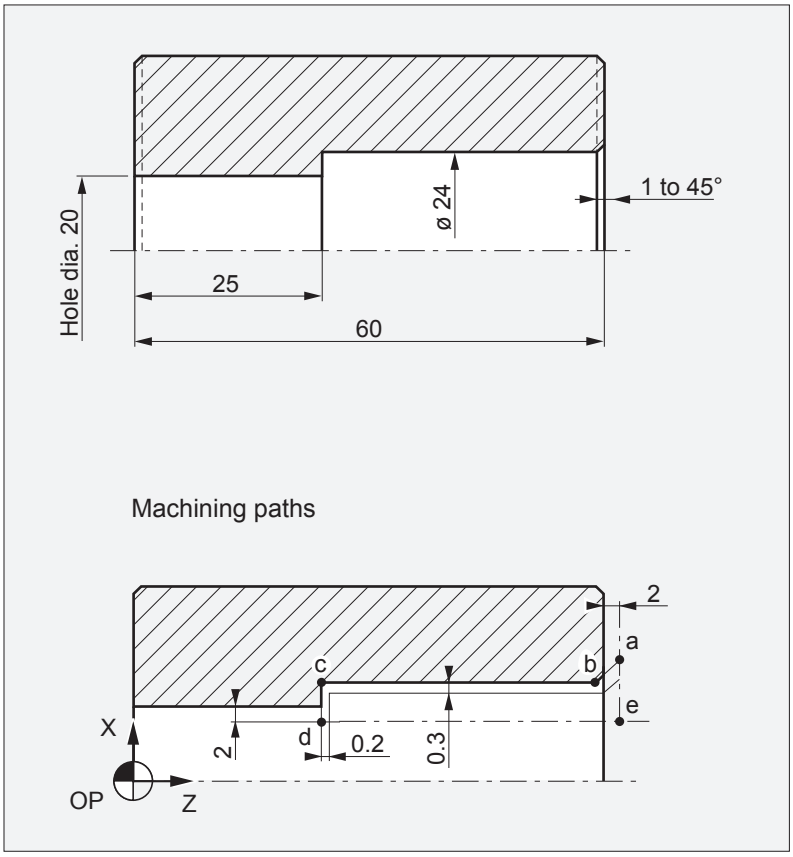
Origin offsets on the Z axis in incremental programming




```
%70
N10
N..
N50
N..
N90
N..
N120 G91 G59 Z-30                    Offset 1
N.. G77 N50 N90                      Machining
N..
N230 G59 Z-30                        Offset 2
N.. G77 N50 N90                      Machining
N..
N350 G91 G59 Z-60 or G90 G59 X0      Cancellation possible with G90 or G91
N..
```

Machining allowance by origin offset

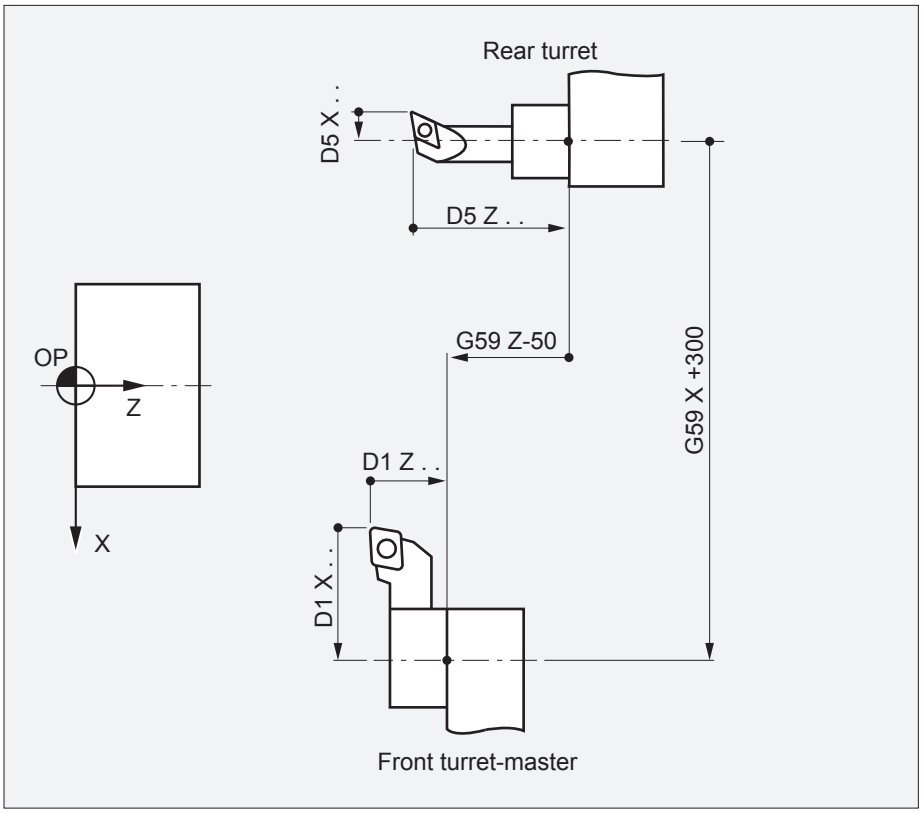
Bore roughing, finishing. Machining allowance: X = 0.3, Z = 0.2




```
N.. ...
N80 G00 G52 X.. Z.. G97 S500
N90 T07 D07 M06 (ROUGHING BORING TOOL)
N100 G95 F0.2
N110 S900 M04 M40
N120 G59 X-0.3 Z0.2                    Origin offsets
N130 G00 G41 X30 Z62                   Point a
N140 G96 S80
N150 G01 X24 Z59                       Point b
N160 Z25                               Point c
N170 X16                               Point d
N180 G00 Z62                           Point e
N190 G59 X0 Z0                         Offsets cancelled
N200 G40 G77 N80 N80
N210 T09 D09 M06 (FINISHING BORING TOOL)
N220 G77 N130 N180                     Finishing
N..
```

Use of origin offset on a lathe with two turrets

On lathes equipped with two turrets (front and rear), the tool dimensions of the secondary turret can be identified with respect to the reference of the master turret by programming an origin offset.



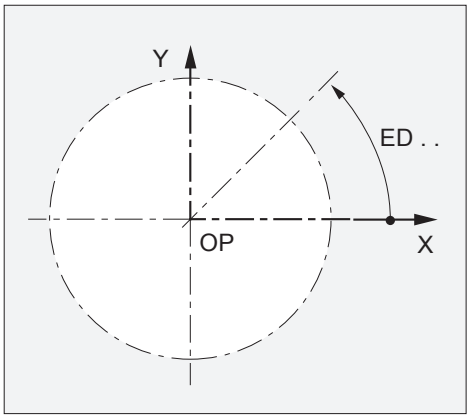

```
% 88
$0 MACHINING WITH FRONT MAIN TURRET
N10 G00 G52 X0 Z0
N20 T01 D01 M06 (EXTERNAL TOOL)
N..
N..
N340 G00 G52 X0 Z0
$0 MACHINING WITH REAR TURRET
N350 T05 D05 M06 (INTERNAL TOOL)
N360 G90 G59 X300 Z-50
N..
N500 G59 X0 Z0
N..
```

5.16.5 Angular offset ► ED..

ED..	Programmed angular offset.

Function ED associated with a value defines an angular rotation with respect to the program origin.

The angular offset applies to the axes of the plane programmed in the blocks following the function.



Syntax

N.. [G90/G91] **ED..**

G90/G91	Programming in absolute or incremental dimensions.
ED..	Value of the angular offset in degrees and thousandths of a degree (format ED+034).

Property of the Function

Function ED is modal.

Cancellation

Angular offset ED.. is cancelled by:

- reprogramming function ED with a zero value (ED0) in absolute dimensions (G90),
- the end of program function (M02),
- a reset.

Notes

Angular offset ED applies to:

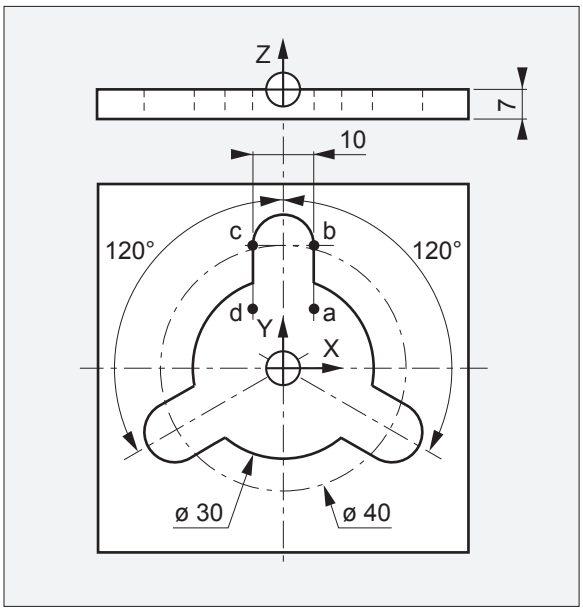
- all the elementary cycles (G81, G45, etc.),
- radius offset (G41, G42),
- PGP (Profile Geometry Programming) unless ED.. is programmed between two blocks that are not completely defined,
- carried or independent secondary axes (U, V, W).

If an angular offset is programmed in a plane (G17, G18 or G19), it is not cancelled by a change of plane. ED.. then applies to the values in the new plane.

Examples

Execution of three grooves in the XY plane

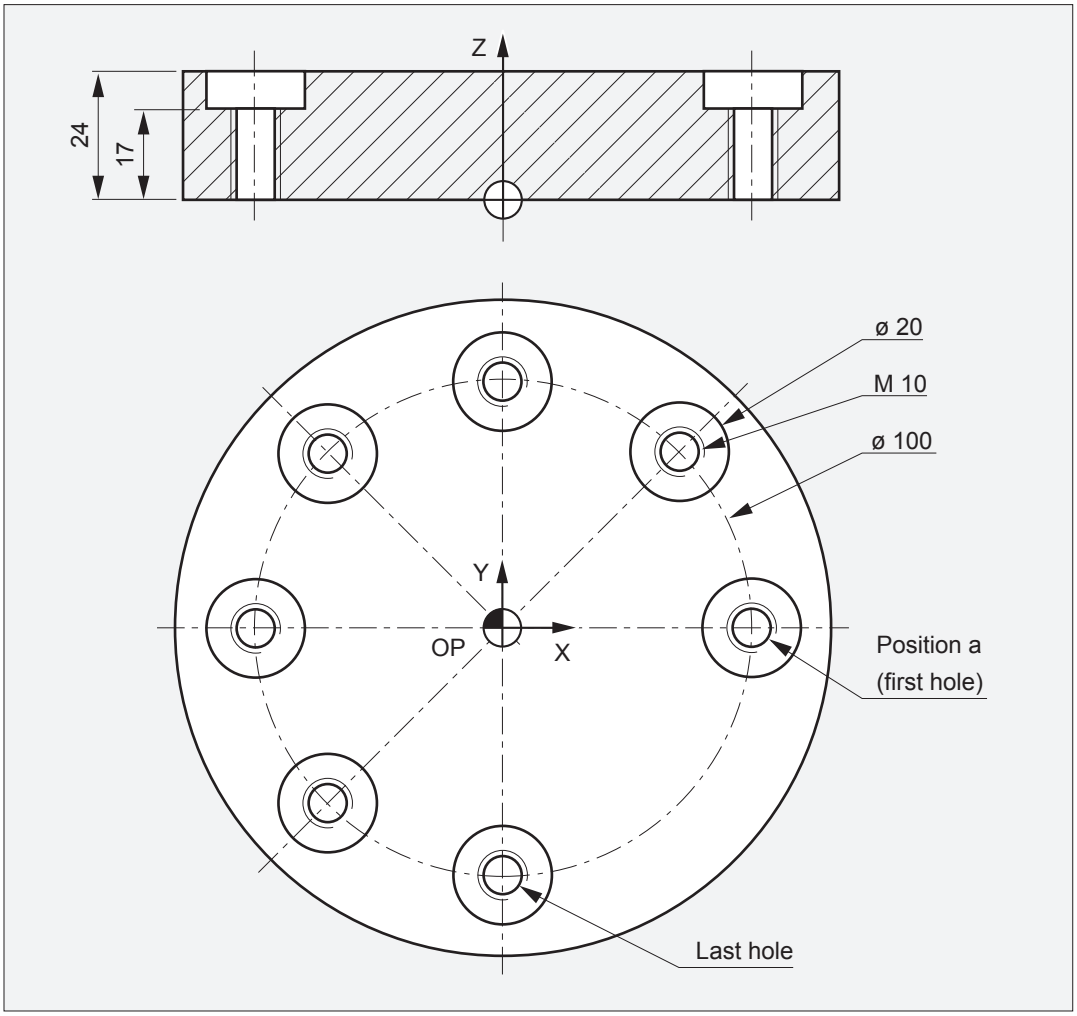
Bore diameter 30 completed.




```
%60
N10 G00 G52 Z-60 M05 M09
$0 3 GROOVES ROUGHING
N20 T06 D06 M06 (CUTTER DIAMETER=8)
N30 S800 M40 M03
N40 G00 G41 X5 Y10                    Point a, groove 1 approach
N50 Z-10                              Approach on Z
N60 G01 Y20 F100 M08                  Point b
N70 G03 X-5 Y20 R5 F50                Point c
N80 G01 Y10 F100                      Point d
N90 ED+120                            Angular offset 120 degrees
N100 G77 N40 N80                      Execution of groove 2
N110 ED+240                           Angular offset 240 degrees
N120 G77 N40 N80                      Execution of groove 3
N130 ED0 G77 N10 N10
$0 3 GROOVES FINISHING
N140 T07 D07 M06 (CUTTER DIAMETER=8)
N150 G77 N30 N130
N160 M02
```

Drilling, spot facing and tapping of seven holes angularly offset

Movement by linear interpolation between holes. Main program




```
%1
$0 MAIN PROGRAM WITH SUBROUTINES %10 %20 %30
N10 G00 G52 Z0 M05 M09
N20 T01 D01 M06 (CUTTER DIAMETER=8.5)
N30 S1000 M40 M03
N40 X50 Y0 Z27                    Point a, approach
N50 G77 H10 S7                    Branch to subroutine %10
N60 G90 ED0                       Offset cancel
N70 G77 N10 N10
N80 T02 D02 M06 (BLADE DIA=20)
N90 S250 M03
N100 X50 Y0 Z27                   Point a, approach
N110 G77 H20 S7                   Branch to subroutine %20
N120 G90 ED0                      Offset cancel
N130 G77 N10 N10
N140 T03 D03 M06 (TAP M10)
N150 S200 M03
N160 X50 Y0 Z27                   Point a, approach
N170 G77 H30 S7                   Branch to subroutine %30
N180 G90 ED0                      Offset cancel
N190 G77 N10 N10
N200 M02
```

Subroutines

```
%10
$0 DRILLING
N10 G90 G81 X50 Y0 Z-5 F90 M08          Drilling cycle
N20 G80 G91 ED45                        Angular offset in incremental dimensions

%20
$0 SPOT FACING
N10 G90 G82 X50 Y0 Z17 EF1 F50 M08      Spot facing cycle
N20 G80 G91 ED45                        Angular offset in incremental dimensions

%30
$0 TAPPING
N10 G90 G84 X50 Y0 Z-8 F300 M08         Tapping cycle
N20 G80 G91 ED45                        Angular offset in incremental dimensions
```

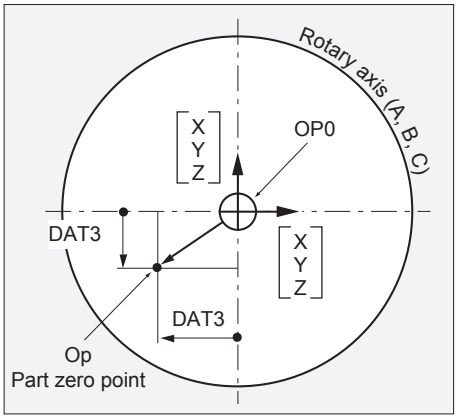
5.16.6 Table Offset by DAT3 with Milling Machine ► DAT3

DAT3	Milling Table offset.

This function applies to rotary axes A, B or C.

The offsets can be applied by entering values:

- on the context "Work Offset”,
- by external parameter E,
- by processor interchange.



Notes

The DAT3 feature takes into account the theoretical shifts caused to a pair of axes when the part zero is not co-incident with the centre of rotation.

Rotary axes and axis pairs affected by DAT3

Rotary axis	Axis pairs
A	YZ
B	ZX
C	XY

The axes of the pair to which a DAT3 can be assigned are identified by stars on the context "Work Offset”. A value entered for an axis not marked by a star is ignored.

Entry of DAT3 values on the NC

Refer to the Flexium HMI Operator Manual M00017.

Entry of DAT3 by external parameters

Parameters E6x004 are use to program the offset (x = logical address of the axis).

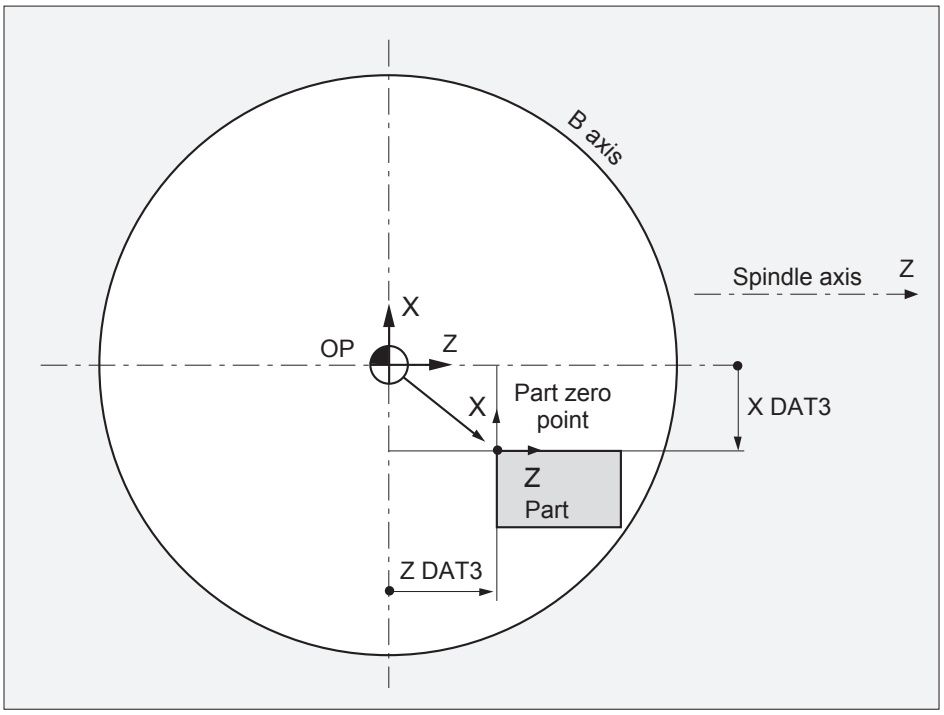
Example

Programming offsets for the B axis affecting the XZ pair.

```
N.. ...
E60004=-50000          Negative 50 mm offset on X
E62004=70000           Positive 70 mm offset on Z
N..
```

The figure below is shown with a slaved B axis (see chapter 4).

The representation is the same for the A and C axes.



5.17 Spline Curve Interpolation ► G48, G06, G49

General

Spline curve interpolation is a mathematical curve fitting method. Spline curves are continuous curves connecting a series of points.

Spline curve interpolation insures continuity of the tangent and a constant acceleration in each of the points specified on the programmed paths.

Machining of a spline curve is programmed by:
- defining the points on the curve,
- specifying a curve execution order.

A spline curve can be inhibited by programming.

Programming

G48	Spline curve definition.

A spline curve definition includes several instructions:
- the definition function,
- the curve number,
- the blocks defining the points on the curve.

G06	Spline curve execution command.

A spline curve execution command is given by a block containing G06 followed by the number of the curve to be executed.

G49	Spline curve deletion.

The system allows memory space to be freed by deleting curves that have already been executed.

A curve is deleted by programming G49 followed by the number of the curve to be deleted.

5.17.1 Spline Curve Interpolation ► G48

G48	Spline curve definition.

Syntax (XY Plane)

N.. **G48** NC.. H../N.. N..

G48	Spline curve definition function.
NC..	Argument defining the curve number.
H..	Number of the subroutine in which are defined the points on the curve (optional).
N.. N..	Numbers of the first and last block defining the points on the curve.

Property of the Function

Function G48 is nonmodal.

Cancellation

Function G48 is cancelled at the end of the block.

Blocks defining the points of a curve

The first and last block defining a curve should include the starting and ending tangents. If the tangents are not known, these blocks must be empty.

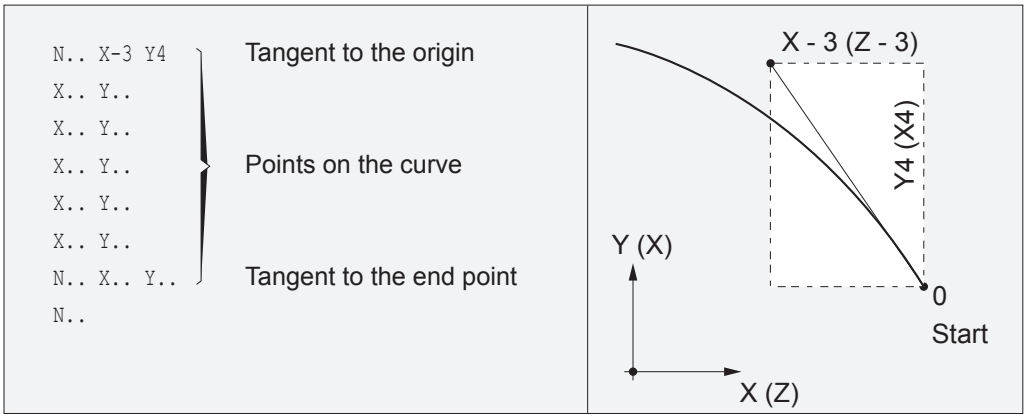
All the definition blocks other than the first and last (tangent to the start and end points) must include points on the curve (no empty lines). Otherwise, the curve would be different from the one desired.

The first point definition block must include all the axes concerned by spline curve interpolation.

If an axis is not programmed in this block, it is not included in spline the curve interpolation even if it is programmed in the following blocks. It is necessary to program the same position in this first block as in the block preceding function G06.

Blocks defining the tangents and points of a spline curve

The projections along the X and Y axes of the tangent to the curve origin have X-3, Y4 as incremental values.



The number of points defining a spline curve is limited to 255. If this value is exceeded, the system returns error message 196.

The points in a spline curve must be defined by ISO programming.

A spline curve must be defined by at least three points. If it is defined by less than three points, the system returns error message 604.

Function G48 must be programmed in state G40, i.e. without radius offset (G41 or G42). Otherwise, the system returns error message 140.

The following functions can be programmed on the curve definition blocks:
- miscellaneous M functions,
- technological F, S or other functions.

Special characters such as $, ( ), etc. should not be programmed in the curve definition blocks.

It is not recommended to program modulo 360 degree axes with spline curves (problem of sign and no movement).

A closed curve is not processed automatically unless the tangents to the start and end points are identical.

Saving spline curve data

The curve coefficients are stored in tables on the program stack. When the program stack is full, the system returns error message 195. In this case, the stack size can be extended (see chapter 8).

The following data are stored for each spline curve defined:
- 1 table with 5 entries,
- 3 tables with P x Q entries, where:
- P: number of axes concerned
- Q: number of points in the profile
- 1 table with P entries.

5.17.2 Spline Curve Execution Command ► G06

G06	Spline curve execution command.

Syntax

N.. **G06** NC..

G06	Function forcing execution of a spline curve.
NC..	Number of the curve to be executed.

Property of the Function

Function G06 is nonmodal.

Cancellation

Function G06 is cancelled at the end of the block.

Notes

The following functions cannot be programmed in the block containing function G06 NC..:

- F : feed rate,
- S : spindle speed,
- T : tool change.

Error Numbers and Messages for Spline Curve

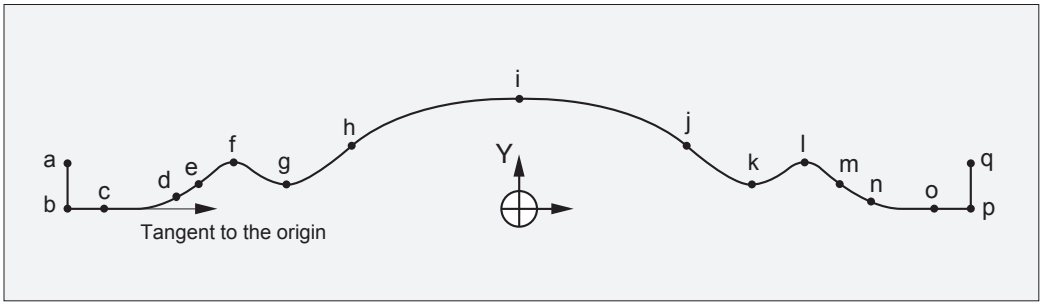
The errors are in error category 600 (see chapter 9).

Examples

Programming of a spline curve defined in the XY plane with a call to definition blocks in the main program

```
%350
G79 N100
N10 X1 Y Z                    Tangent to the origin
X-240 Y0 Z0 F300              Start of the curve and definition of the
X-200 Y10 Z5 (c)              axes to be interpolated
X-180 Y20 Z10
X-160 Y30 Z15
X-140 Y20 Z20
X-80 Y40 Z35
X0 Y60 Z40                    Points on the curve (c to n)
X80 Y40 Z35
X140 Y20 Z20
X160 Y30 Z15
X180 Y20 Z10
X200 Y10 Z5
X240 Y0 Z0 (n)
N20                           No tangent in the end point
N100 T3 D3 M6 (TOOL R5)
N110 S3000 M40 M3
N120 G48 NC5 N10 N20          Definition of the curve
N130 G0 X-300 Y20             Point a
N140 Z0
N150 G1 G41 Y0                Point b
N160 X-240                    Point c
N170 G06 NC5                  Curve execution command
N180 G1 X300                  Point p
N190 G0 G40 Y20               Point q
N200 G0 G52 Z0 M5
N210 M2
```

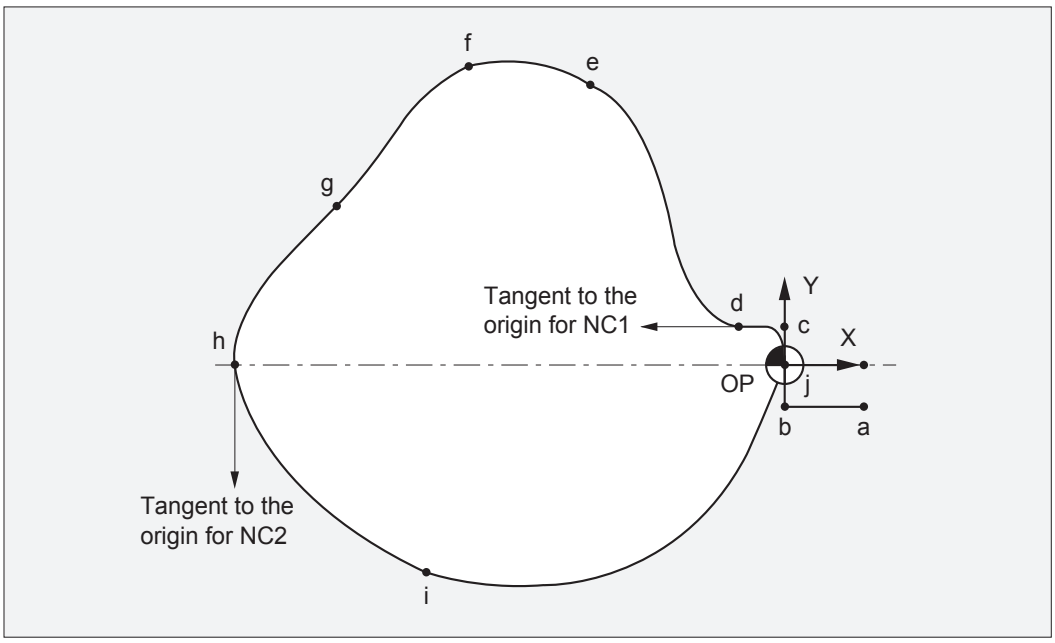
Representation of machining



Programming of two consecutive spline curves defined in the XY plane with call to definition blocks in two subroutines

```
%300
N10 G0 G52 Z0
N20 T1 D1 M6 (CUTTER R5)
N30 S2000 M40 M3
N40 G92 R1
N50 G48 NC1 H31 N10 N70          Curve 1 definition
N60 G42 X20 Y-10                 Point a
N70 Z-3
N80 G1 X0 F200                   Point b
N90 EA90 EB5 ES-                 Point c
N100 EA180 X-13 Y10              Point d
N110 G06 NC1                     Curve 1 execution command
N120 G48 NC2 H32 N10 N50         Curve 2 definition
N130 G06 NC2                     Curve execution command
N140 G0 G40 X20
N150 G52 Z0 M05
N160 M02

%31
N10 X-1 Y0                       Tangent to the origin
N20 X-13 Y10
N30 X-50 Y70                     Points on curve 1 (d, e, f, g, h)
N40 X-78 Y75
N50 X-104 Y55
N60 X-140 Y0
N70 X0 Y-1                       Tangent to the end point

%32
N10 X0 Y-1                       Tangent to the origin
N20 X-140 Y0
N30 X-82 Y-60                    Points on curve 2 (h, i, j)
N40 X0 Y0
N50 X1 Y5                        Tangent to the end point
```

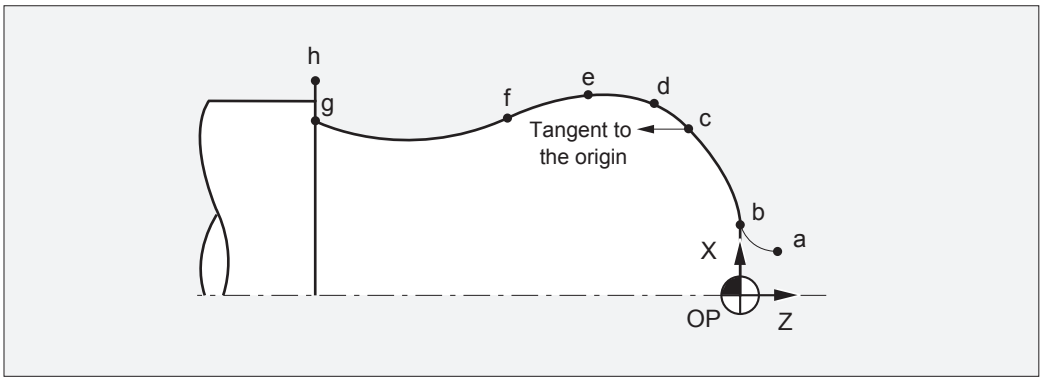
Representation of machining



Programming of a spline curve defined in the ZX plane with a call to definition blocks in the main program

```
%70
N10 ...
N..
N..
$0 FINISHING OF THE SPLINE PROFILE
N180 G0 G52 X0 Z0 G97 S900 M05
N190 G79 N220                          Jump to block N220
N200 X0 Z-1                            Tangent to the origin of the curve
G01 X35 Z-10 F0.15
X37 Z-13
X40 Z-25                               Points in the curve (c, d, e, f, g)
X35 Z-50
X40 Z-80
N210                                   No tangent to the end point
N220 T01 D01 M06 (FINISHING TOOL)
N230 S900 M03 M40
E11005=0                               Forcing programming on radius
N240 G48 NC1 N200 N210                 Spline curve definition
N250 G0 G42 X10 Z5                     Point a
N260 G96 S300
N270 G95 F0.3
N280 G02 X15 Z0 I15 K5                 Point b
N290 G03 X35 Z-10 R30                  Point c
N300 G06 NC1                           Curve execution order
N310 G01 G40 X45                       Point h
E11005=1                               Forcing programming on diameter
N320 G77 N180 N180
N330 M02
```

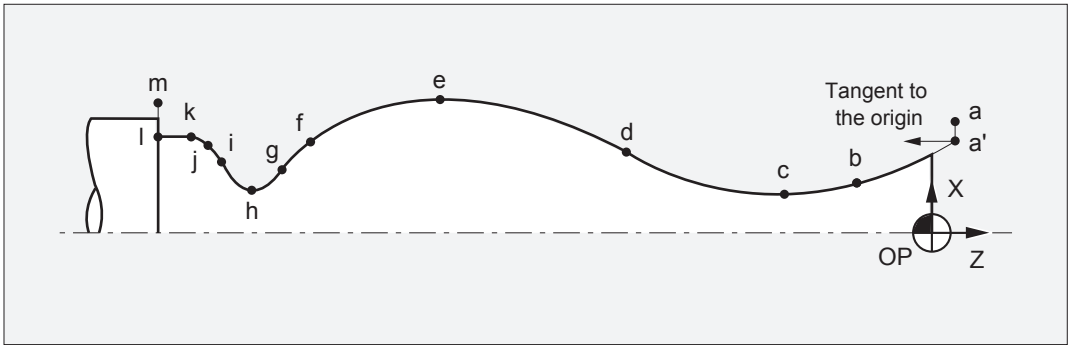
Representation of machining



Programming of a spline curve defined in the XZ plane with call to the definition blocks in a subroutine

```
%150
N10
N..
N200 G0 G52 X0 Z0 G97 S500 M05
N210 T03 D03 M06 (FINISHING TOOL)
N220 S2000 M40 M04
E11005=0                    Forcing of programming with reference to the radius
N230 G48 NC1 H151 N10 N130  Curve definition
E11005=1                    Programming with reference to the diameter
N240 G0 X30 Z5              Point a
N250 G92 S3000
N260 G96 S250
E11005=0                    Forcing of programming with reference to the radius
N270 G01 G42 X11 G95 F0.3   Point a'
N280 G06 NC1                Execution order
E11005=1                    Programming with reference to the diameter
N290 G01 Z-105              Point l
N300 G40 X30                Point m
N310 G77 N200 N200
N320 M02

%151
N10 X0 Z-1                  Tangent to the origin
N20 X11 Z5 F0.1             Curve start
N30 X6 Z-10                 }
N40 X5 Z-20                 }
N50 X12 Z-40                }
N60 X17 Z-60                }
N70 X12 Z-80                } Points on the curve (a to k)
N80 X8 Z-84                 }
N90 X5 Z-88                 }
N100 X9 Z-91                }
N110 X11 Z-93               }
N120 X12 Z-95               }
N130 X0 Z-1                 Tangent to the end point
```

Representation of machining



5.17.3 Freeing Memory by Deleting a Spline Curve ► G49

G49	Spline curve deletion.

This function is used to free the memory space occupied by curves already executed.

Syntax

N.. **G49** NC..

G49	Spline curve deletion.
NC..	Number of the curve to be deleted.

Property of the Function

Function G49 is nonmodal.

Cancellation

Function G49 is cancelled at the end of the block.

Notes

Function G49 must be programmed in state G40, i.e. without radius offset (G41 or G42).

Otherwise the system returns error message 140.

Example

Definition and execution of a spline curve followed by deletion of the curve

```
%300
N10 ...
N.. G79 N200
N110 X.. Y..                    } Points in curve 1
N..                             }
N..                             }
N150 X.. Y..                    }
N..
N200 G48 NC1 N110 N150          Curve 1 definition
N300 G06 NC1                    Curve 1 execution
N..
N..
N450 G49 NC1                    Curve 1 deletion
N..
```

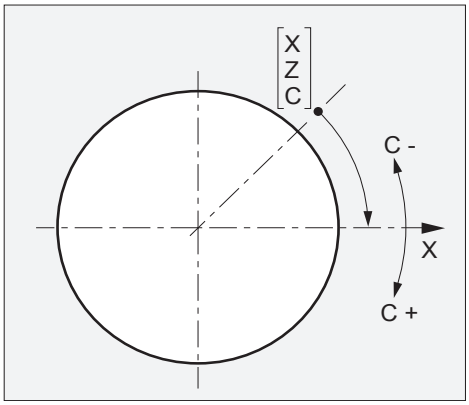
5.18 Coordinates Systems with C Axis with Turning Machine

5.18.1 Programming in Polar Coordinates ► G20

G20	Programming in polar coordinates (X, Z, C).

This function is used to program the linear axes X and Z and control a rotary axis C, modulo 360 degrees.

X/Z programming is the same as programming a system without a C axis.



N.. [G40] **G20** [G00/G01][X.. Z.. C..][F..]

G40	Radius offset cancel.
G20	Programming on axes X, Z and C (angular value C.. with reference to the origin).
G00/G01	Interpolations permitted.
X.. Z.. C..	End point.
F..	Feed rate.

Properties of the Function

Function G20 is modal and is the default function.

Cancellation

Function G20 is cancelled by functions G21 and G22.

Notes

Programming of function ER is excluded in G20.

Notes on declaration of the C axis

Control of the C axis requires the use of:
- a servo-drive
- a position encoder
- a special drive arrangement.

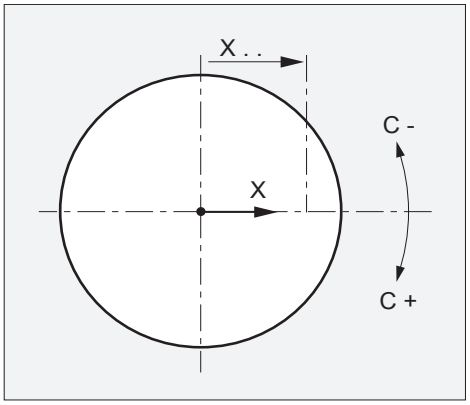
Example

```
N.. ...
N1050 G40 G21 (or G22) ...          Machining in cartesian or cylindrical coordinates
N..
N1220 G40 G20 G01 X.. Z..           Return to initial state
N..
```

5.18.2 Definition of the Start X for Interpolation on the C Axis ► G98

G98	Definition of the start X for interpolation on the C axis.

When the system is in G20 mode, this function is used to compute the feed rate on the C axis in mm/min.



Syntax

N.. **G98** X..

G98	Definition of the start X for interpolation on the C axis.
X..	Start X.

Property of the Function

Function G98 is nonmodal.

Cancellation

Function G98 is cancelled at the end of the block.

Notes

Programming G98 X.. does not cause any movement.

When function G98 X.. is not programmed, the last value declared with the X axis is used for the computation.

When the C and X axes are interpolated simultaneously, the system performs its feedrate calculation from the mean radius.

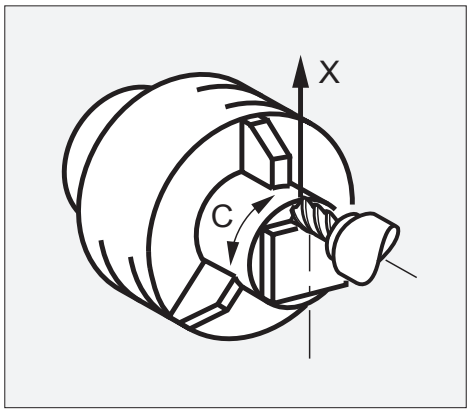
5.18.3 Programming in Cartesian Coordinates ► G21

G21	Programming in cartesian coordinates (X, Y, Z).

The system performs cartesian/polar coordinate conversion (X-Y converted to X-C).

Interpolation of the X and C axes allows milling in the plane perpendicular to the spindle axis.

The tool is driven by an auxiliary spindle·



Syntax

N.. [G40] **G21** [G00/G01] [G41/G42] [X.. Y.. Z..] [G94 F..]

G40	Radius offset cancel.
G21	Programming in cartesian coordinates.
G00/G01	Interpolations.
G41/G42	Radius offset to the left or right of the profile.
X.. Y.. Z..	End point.
G94 F..	Feed rate in mm/min.

Property of the Function

Function G21 is modal.

Cancellation

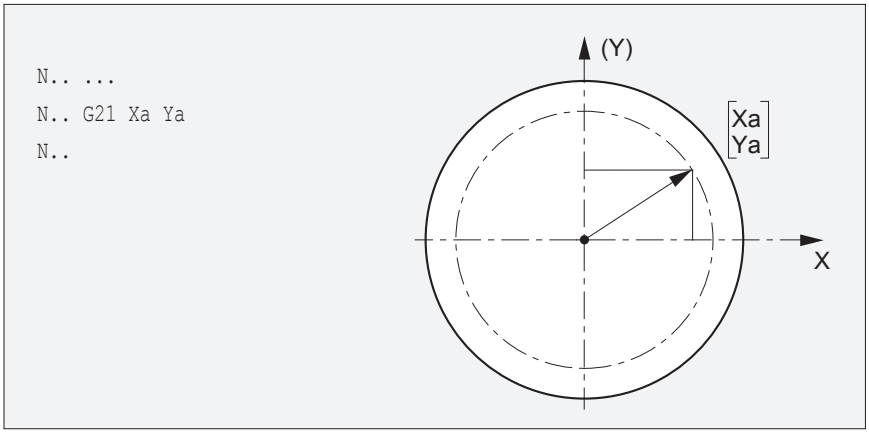
Function G21 is cancelled by function G20.

Programming

Function G21 cannot be programmed during a canned cycle. The use of a cycle causes generation of error message 76.

G21 is used for programming the X, Y and Z addresses by linear interpolation (the Z address is incompatible with circular interpolation G02 and G03).

Programming a point in G21:



During transition from G20 to G21:
- the system must be in G40 mode (radius offset cancel),
- if the last block in G20 includes an X demand, it must have a positive value,
- the system must not be programmed in PGP (Profile Geometry Programming).

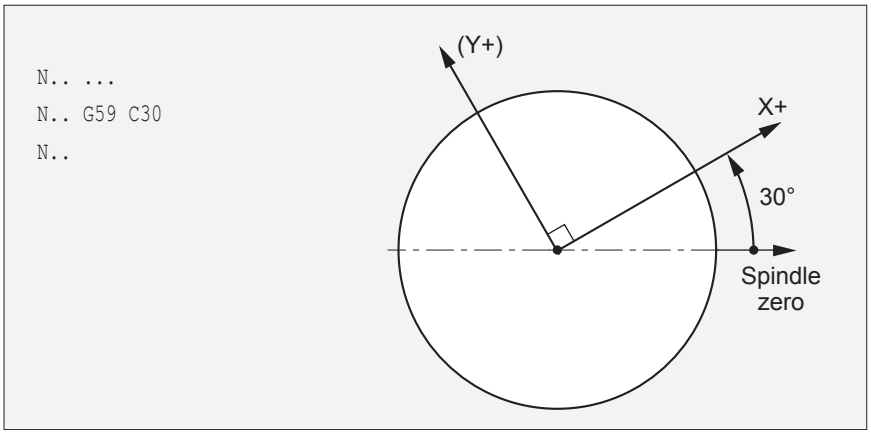
During cancellation of G21 by G20, the system must be in G40 mode (radius offset cancel).

Failure to comply with the transition and cancellation conditions causes generation of error message 75.

Offsets

Programming of offsets is allowed:
- G59 X.. Y.. or G59 C.. : Programmed origin offset,
- ED.. : Programmed angular offset,
- G51.. :Mirroring.

An angular offset programmed on the C axis causes a rotation of the X and Y coordinates, e.g.:



Tool corrections

No change of tool corrections are authorised in G21. The corrections must be programmed and modified in G20.

Feed rate limiting

When the tool is moving toward the centre of the part, the constant feed rate along the path is achieved by increasing the angular feed rate on the C axis and decreasing the linear feed rate on the X axis.

The system may therefore have to decrease the tangential machining feed rate so as not to exceed the maximum rotation speed on the C axis when the distance to the center decreases.

However as this feed rate cannot be decreased below 1 inc/scan. the system will issue an error (error34) when the minimum radius is reached.

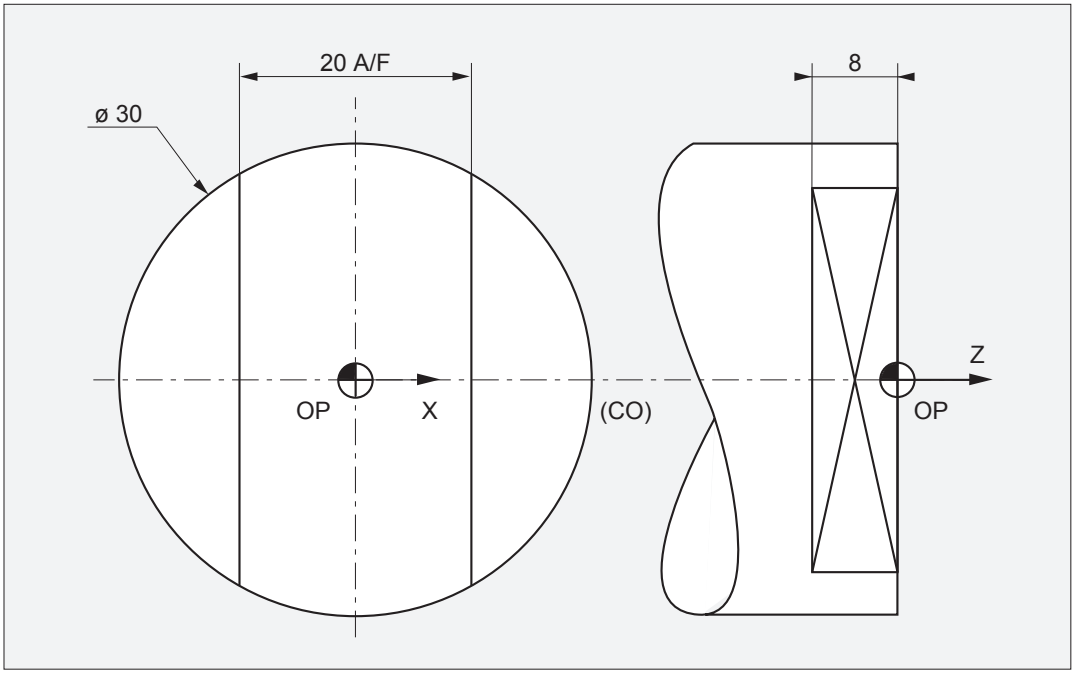
The minimum radius R is calculated by the following equation:

$$R = \frac{3440000}{Vc\ Max * Sampling\ period}$$

Where:
- R is the distance between the cutter centre and the C axis centre in µm.
- VCmax is the maximum speed on the C axis expressed in degrees/minute (see machine parameter P30).
- Sampling period : Axis scanning period in ms.
  (see Flexium Parameters Manual M00021 machine parameter P50).

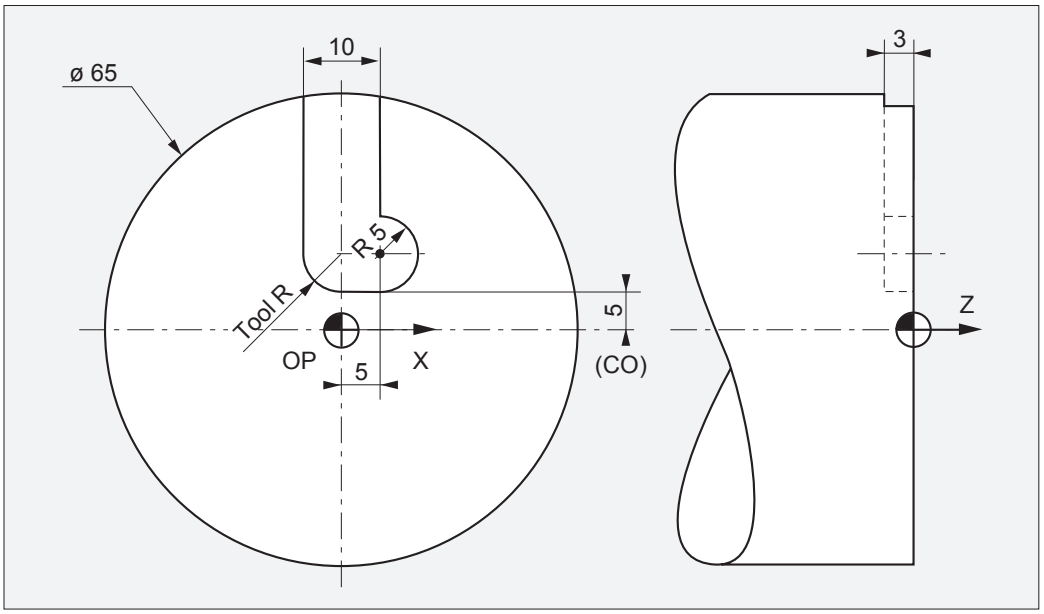
Examples

Execution of two flats




```
N..
N200 G00 G40 G52 X.. Z.. M05
N210 T12 D12 M06 (CUTTER D=10)
N220 G97 S1000 M.. $0 AUXILIARY SPINDLE ROTATION
N230 X50 Z2                                    Tool approach along XZ
N240 G94 F..
N250 G21 G01 G42 X10 Y11.18
N260 G00 Z-8                                   Pass setting on Z
N270 G01 Y-11.18                               Execution of flat 1
N280 G00 X-10
N290 G01 Y11.18                                Execution of flat 2
N300 G00 Z3
N310 G40 G20 X100
N320 ...
```

Execution of a shape




```
N..
N200 G00 G40 G52 X.. Z.. M05
N210 T8 D8 M06 (CUTTER D=8)
N220 G97 S1200 M.. $0 AUXILIARY SPINDLE ROTATION
N230 X70 Z2                                    Tool approach along XZ
N240 G94 F..
N250 G92 R1
N260 G21 G00 G41 X-5 Y35
N270 G00 Z-3                                   Pass setting on Z
N280 G01 Y5
N290 X5
N300 G03 X5 Y15 R5
N310 G01 Y35
N320 G00 G40 G20 X200
N330 G52 X0 Z0
N340 ...
```

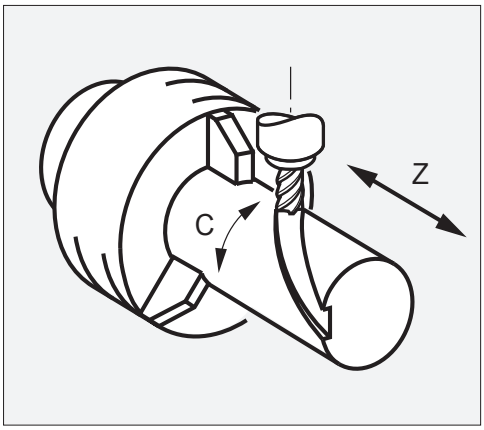
5.18.4 Programming in Cylindrical Coordinates ► G22

G22	Programming in cylindrical coordinates (X, Y, Z).

The system performs cylindrical/polar coordinate conversion (Y-Z converted to Z-C).

Interpolation of the C axis allows milling on the developed cylinder with radius X.

The tool is driven by an auxiliary spindle.



Syntax

N.. [G40] **G22** [G00/G01] [G41/G42] [X.. Y.. Z..] [G94 F..]

G40	Radius offset cancel.
G22	Programming in cylindrical/polar coordinates.
G00/G01	Interpolations.
G41/G42	Radius offset to the left or right of the profile.
X.. Y.. Z..	End point.
G94 F..	Feed rate in mm/min.

Property of the Function

Function G22 is modal.

Cancellation

Function G22 is cancelled by function G20.

Programming

G22 is used for programming addresses:
- X, Y and Z by interpolation G01
- Y, Z, J and K by interpolations G02 and G03.

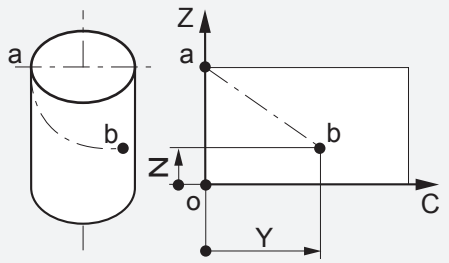
When the C axis rotates in the positive direction, the Y axis also increases positively.

Position Y0 coincides with position C0 of the C axis.

With G22, canned cycles are not authorised. The use of a canned cycle causes generation of error message 76.

Programming of a point in G22:

```
N.. ...
N.. Xa Za
N..
N.. G22 Yb Zb
N..
```



During transition from G20 to G22:
- the system must be in G40 mode (radius offset cancel),
- if the last block in G20 included a movement along X, it must be programmed with a positive value,
- the system must not be programmed in PGP (Profile Geometry Programming).

During cancellation of G22 by G20:
- the system must be in G40 mode (radius offset cancel),
- the system must not be programmed in PGP.

Failure to comply with the transition and cancellation conditions causes generation of error message 75.

Offsets

No offsets are allowed in G22 mode.

In G20 mode, origin offsets G59 are possible:

- Before the call to function G22, the start point on the C axis can be modified, e.g. N.. G59 C..
- During the program, the position of the Z axis can be modified, but this requires a return to state G20, e.g. N.. G59 Z..

Tool corrections

No changes in the tool corrections are authorised in G22. The corrections must therefore be programmed and modified in G20.

Feed rate limiting

The feed rate is limited by the maximum feed rate authorised on the C axis and the current value of the positioning radius.

For machining small diameters, the system may have to decrease the feed rate so as not to exceed the maximum rotation speed on the C axis.

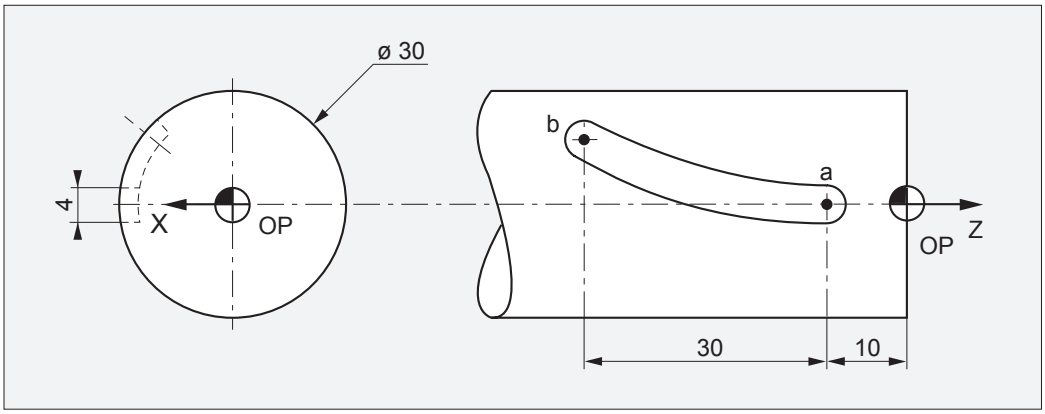
The system automatically limits the minimum feed rate when it is incompatible on the programmed path. When the minimum speed is reached, the system returns error message 34.

Initialisation of the cartesian position on the Y axis

Positioning to the first point programmed in G22 is carried out by computing the function start value on the Y axis from the value on the X axis and on the C axis, interpreted between -180 and +180, giving:
Y = θ . X (θ = angle on the C axis between + and - π).

Example

Application




```
N..
N200 G00 G40 G52 X.. Z.. M05
N210 T08 D08 M06 (CUTTER D=4)
N220 G97 S2500 M.. $0 AUXILIARY SPINDLE ROTATION
N230 X32 Z-10                                    Point a, tool approach along XZ
N240 C0                                          C axis in zero point
N240 G94 F50
N250 G01 X28                                     Cut application in X
N260 G22 Y-10 Z-40 F..                           Point b
N270 G20 X32
N280...
```

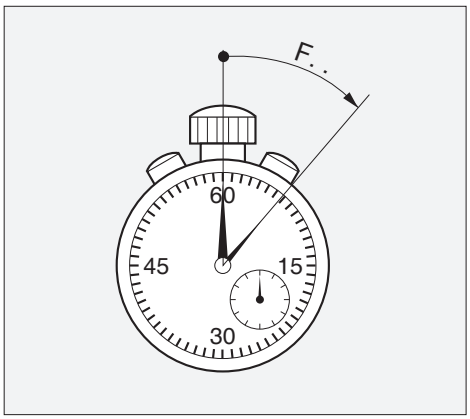
Homing is required on the C axis.

5.19 Other Functions

5.19.1 Dwell ► G04

G04	Programmable dwell.

Continuation of the program is interrupted for the time programmed with argument F.



Syntax

N.. **G04** F..

G04	Programmable dwell.
F..	Dwell in seconds (0.01 to 99.99 seconds, format F022). Mandatory argument F must be programmed immediately after the function.

Property of the Function

Function G04 is nonmodal.

Cancellation

Function G04 is cancelled at the end of the block.

Function G04 may be cancelled before the end of the block when it is programmed with function G10.

Notes

Function G04 F.. does not cancel the feed rates programmed with F in the previous block(s).

Example:

```
N.. ...
N50 G01 Xa Ya F200
N60 G04 F1.5
N70 Xb Yb
N..
```

Feed rate 200 mm/min
Dwell 1.5 seconds
After 1.5 seconds machining resumes at 200 mm/min

If G04 is programmed at the start of a block containing a move, the dwell will occur at the end of the block.

Example:

```
N.. ...
N.. G01 G04 F5 X100 Y100
N..
```

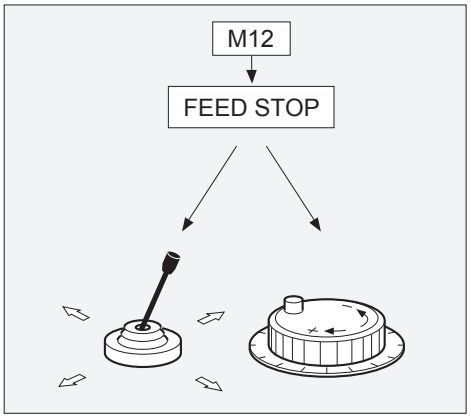
Dwell after movement

5.19.2 Programmed Feed Stop ► M12

M12	Programmed feed stop.

This function requires operator action after a cycle stop action.

The axis jogs or handwheel are made available.



Syntax

N.. **M12** [$0...]

M12	Programmed feed stop.
$0...	Message for the operator (see chapter 5.19).

Properties of the Function

Function M12 is a decoded nonmodal «post» function.

Cancellation

The function is cancelled by pressing the CYCLE key on the machine panel.

Notes

M12 should be activated by the machine builder to work as explained.
(see Machine Operation manual).

When function M12 enables the axis jogs or handwheel:

- the operator can only make movements in continuous jog mode (J.ILL),
- during axis movements, the system remains in the current execution mode: automatic (AUTO) or single step (SINGLE).

When the operator cancels the function after any axis movements, the program is resumed from the new position (axis recall not required).

Function M12 is ignored in the test (TEST) and sequence number search mode (SEARCH).

Example

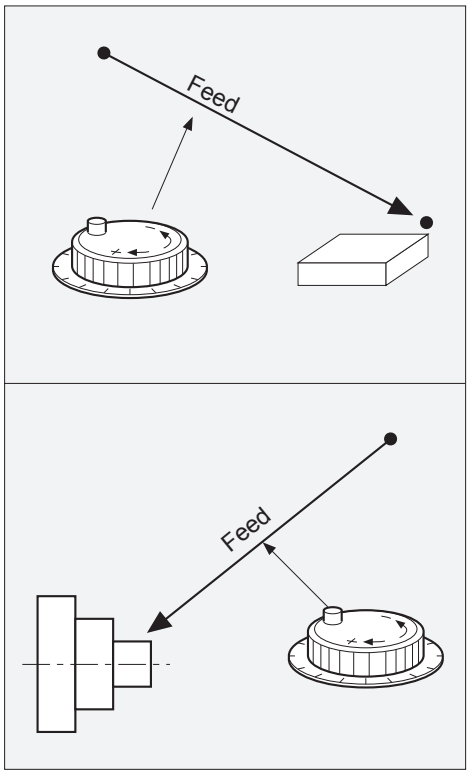
```
N.. ...
N300 G00 Z80 M12 $0 MOVEMENT ON X, THEN CYCLE START
N310 ..
N..
```

5.19.3 Feed Increase ► G12

G12	Feed Increase by handwheel.

When the machine is equipped with handwheels, this functions is used to increase the speed of movement on the linear or circular paths programmed in the block.

Overspeed is applied to the first handwheel.



Syntax

N.. [G01/G02/G03] **G12** X.. Y.. Z.. [F..] [$0 ...]

G01/G02/G03	Linear or circular interpolation.
G12	Enabling of rapid feed by handwheel.
X.. Y.. Z..	End point.
F..	Feed rate.
$0 ...	Message for the operator (see chapter 5.19).

Property of the Function

Function G12 is nonmodal.

Cancellation

Function G12 is cancelled at the end of the block.

Notes

Continuation to the next block is carried out when the programmed position is reached.

The overspeed coefficient applied by function G12 is defined by the OEM.

Example

```
N.. ...
N60 G01 G12 X.. Y.. F200 $0 ACTUATE THE HANDWHEEL
N..
```

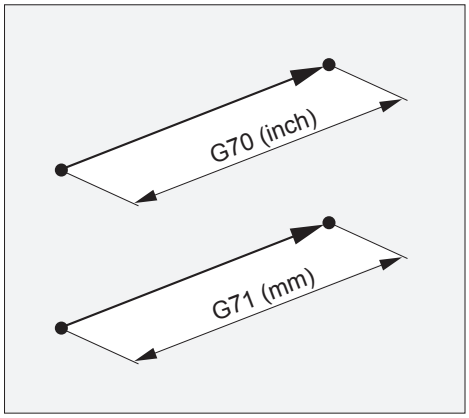
5.19.4 Programming in Inches or Metric Data ► G70, G71

G70	Inch data input

This function is used to program data in inches.

G71	Metric data input.

This function is used to program data in metric units.



Syntax

N.. **G70/G71**

G70	Inch data input.
G71	Metric data input.

Properties of the Functions

Functions G70 and G71 are modal.

Function G71 is the default function.

Cancellation

Functions G70 and G71 cancel one another.

Notes

Programming can be changed from inches to metric units and vice versa in machine parameter P7 (see Flexium Parameters Manual M00021).

It should be noted that the choice of the displayed unit (inches or metric units) is made by PLC variable "Flexium_NCK.W_ALLCNC.InchUnit".

Formats specific to inch data input (G70)

Addresses	Formats
X, Y, Z, U, V, W, I, J, K	+044
P, Q, R, ER etc...	+044

Inch data input (G70) for program variable «L» and external parameters «E»

The programming must be suited to the operations carried out on the dimensions and external parameters «E» expressed in their own units. See chapter 7.

Examples

Programming of a dimension using program variable L

```
N.. ...
N.. G70
N.. L1 = 10
N.. G01 XL1                    L1 is equal to 10 inches
```

Programming of a dimension using an external parameter E

```
N.. ...
N.. G70
N.. E80000 = 100000
N.. G01 XE80000                E80000 is equal to 10 inches
                               (format 044)
```

Modification of a tool dimension by the program

External parameter E50001 represents the length of tool 1.

The value stored into the "E" parameter is held in NC internal measurement unit (Refer to P4N2 setting on M00021 Parameters manual).

The coefficient must be adjusted in relation to P4N2 setting "NC internal measurement unit". In case of µm the coefficient is 25400, in case of 1/10µm the coefficient is 254000.

```
N.. ...
N.. G70
N.. L10 = E50001/25400         L10 = length of tool 1 converted to inches
N.. L2 = 3 + L10               L2 = length of tool 1 + 3 inches
N.. E50001 = L2 * 25400
```

Notes

In the example above it is assumed that the internal resolution of the system is 1µm.

The value 25400 needs to be adapted fort other resolutions (e.g. 254000 for 1/10µm resolution).

The system internal resolution is defined by the machine builder in parameter P42 N2.

5.19.5 Axis Clamping and Unclamping ► M10, M11

M10	Clamp.

This function is used to clamp those axes with clamps that are not required to move.

M11	Unclamp.

This functions cancels axis clamping.

Syntax

N.. [G00/G01/G02/G03] **M10/M11** X.. Y.. Z.. A.. B.. C..

G00/G01/G02/G03	Linear or circular interpolation.
M10	Clamp.
M11	Unclamp.
X.. Y.. Z.. A.. B.. C..	End point.

Properties of the Functions

Function M10 is a decoded modal «post» function.

Function M11 is a decoded modal «pre» function.

Cancellation

Functions M10 and M11 cancel one another.

Notes

The axes recognised as able to be clamped by function M10 are declared by the machine builder.

When function M10 is programmed, the system generates a timeout followed by a wait for a acknowledgement before executing the movements in the next block if some axes declared as clampable change status (will start or stop).

After a reset, function M10 is initialised on the channels which have axes that can be clamped.

Example

```
N.. ...
N40 M10                          Working with clamps enabled.
N50 G00 X.. Y.. Z.. B..          The X, Y, Z, B axes are not clamped.
N60 G01 X.. Y.. Z.. F200         The B axis which does not move is clamped.
N..
N210 M11                         Unclamp all axes.
N..
```

5.19.6 Coolant ► M08, M07, M09

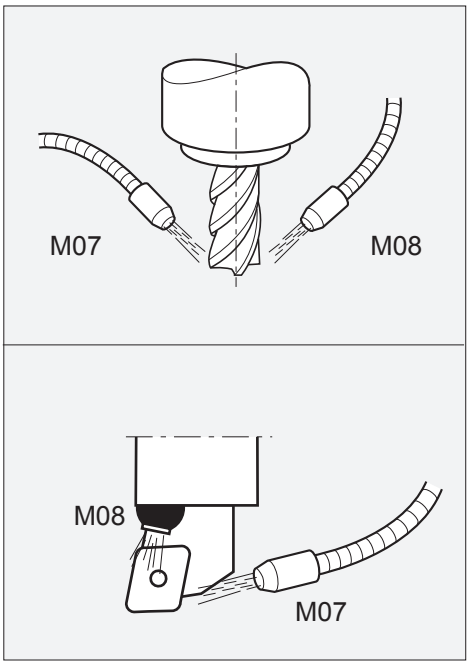
M08	Coolant 1 on.

M07	Coolant 2 on.

These functions turn on the coolant pumps.

M09	Coolant off.

This function stops the coolant pumps.



Syntax

N.. **M08/M07**	
M08	Coolant 1 on.
M07	Coolant 2 on.

N.. **M09**	
M09	Coolants 1 and 2 off.

Properties of the Functions

Functions M07 and M08 are decoded modal «pre» functions.

Function M09 is a decoded modal «post» function. It is the default function.

Cancellation

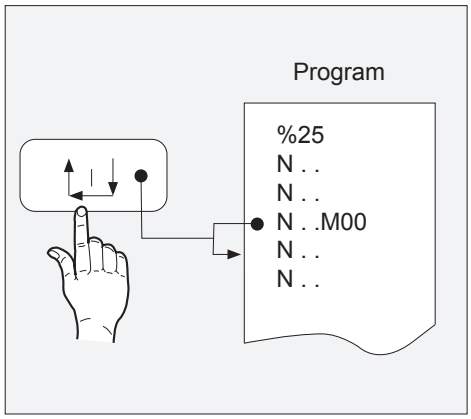
Functions M08 and M07 are cancelled by functions M09 or M02.

Example

```
N.. ...
N40 G00 X.. Y.. M08                Coolant 1 on
N50 G01 Z.. M07                    Coolant 2 on
N..
N230 G00 G52 Z-100 M05 M09         All Coolant off
N..
```

5.19.7 Program Stop ► M00

M00	Program stop.



This function stops execution of the current program.

After intervention or a check, the operator restarts the program.

Syntax

N.. [G40] **M00** [$0 ...]

G40	Radius offset cancel.
M00	Program stop.
$0 ...	Message for the operator (see chapter 5.19).

Properties of the Function

Function M00 is a decoded nonmodal «post» function.

Cancellation

This function is cancelled by pressing the CYCLE key.

Notes

When function M00 is read in a block:
- continuation to the next block is interrupted and spindle rotation is stopped,
- the contents of the field in the status window are modified and the indicator CYCLE is replaced by M00.

After action or check by the operator, pressing the CYCLE key restarts the programme.
The indicator M00 is replaced by CYCLE.
M00 must be programmed with the system in G40 mode (radius offset cancel).
Spindle rotation must be reprogrammed after a stop programmed by M00.

Transfer of function M00 to the PLC

Function M00 is transmitted to the PLC at the end of execution of the block in which it is programmed, but before waiting for channel synchronisation if required (G78..) in a system with multiple channels (in this case, stopping of the spindle is taken into account when synchronisation is completed).

Example

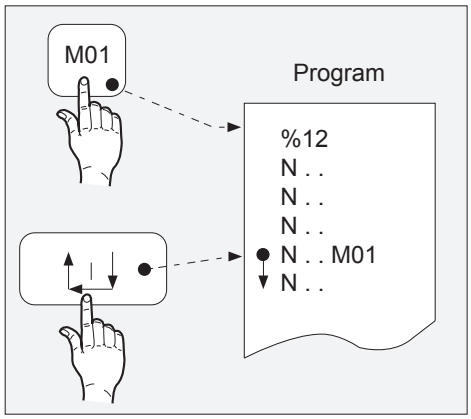
```
N.. ...
N.. G01 G41 X.. Y.. F200 M08
N..
N190 G00 G40 Z200 M09                  Tool retraction before intervention
N200 M00 $0 REMOVE CHIPS BEFORE        Stop and message
FINISHING
N210 G00 G41 X.. Y.. Z.. M03           Continuation of the program
N..
```

5.19.8 Optional Stop ► M01

M01	Optional stop.

When confirmed by the operator M01 causes a stop in program execution.

After intervention or check, the cycle is restarted by the operator.



Syntax

N.. [G40] **M01** [$0 ...]

G40	Radius offset cancel.
M01	Optional stop.
$0 ...	Message for the operator (see chapter 5.19).

Properties of the Function

Function M01 is a decoded nonmodal «post» function.

Cancellation

The function is cancelled by pressing the CYCLE key.

Notes

When function M01 is read in a block (M01 confirmed):
- continuation to the next block is interrupted and spindle rotation is stopped,
- the contents of the field in the status window are modified and the indicator CYCLE is replaced by M00.

After action or check by the operator, pressing the CYCLE key restarts the program. The indicator M01 is replaced by CYCLE.
M01 must be programmed with the system in G40 mode (radius offset cancel).
Spindle rotation must be reprogrammed after a stop programmed by M01.

Transfer of function M01 to the PLC

Function M01 is transmitted to the PLC at the end of execution of the block in which it is programmed, but before waiting for channel synchronisation if required (G78..) in a system with multiple channels (in this case, stopping of the spindle is taken into account when synchronisation is completed).

Example

```
N.. ...
N.. G01 G42 X.. Y.. F200 M08
N..
N290 G00 G40 Z200 M09                Tool retraction before intervention
N300 M01 $0 CHECK DIMENSION 50       Stop if M01 is confirmed
EVERY 5 PARTS                        and message
N310 G00 G42 X.. Y.. M03             Continuation of the program
N..
```

5.19.9 Inhibition of MDI Modes ► M999, M998

M999	Programmed inhibition of the manual data input (MDI) mode and subroutine calls by the PLC.

As long as this function is active the operator cannot call the MDI mode and the PLC cannot call subroutines.

M998	Reactivation of the manual data input (MDI) mode and subroutine calls by the PLC.

Syntax

N.. **M998/M999**

M999	Programmed cancellation of the manual data input (MDI) mode and subroutine calls by the PLC function.
M998	Reactivation of the manual data input (MDI) mode and subroutine calls by the automatic control function.

Properties of the Functions

Functions M998 and M999 are modal «pre» functions.

Function M998 is the default function.

Cancellation

Functions M998 and M999 cancel one another (M998 and M999 can also be cancelled by M997 and M02).

M999 is cancelled by a reset.

Notes

Programming M999 allows:
- manual movements (JOG or MANUAL),
- use of enhanced feed by handwheel (G12),
- use of stop on switch (G10).

Programming M999 inhibits:
- switch to sequence number search (SEARCH) mode during execution of a sequence of blocks «masked» by programming M999 (ignored by the NC),
- action by the PLC or the operator during execution of a sequence of blocks.

The goal of M999 is to authorize writing into variables L100 to L199, L900 to L999 and symbolic variables without holding block preparation.

In M998 status each attempt to write into these variables will result in stopping block preparation and therefore preventing cutter compensation.

Detailed information can be found in the Flexium Extended Programming manual M00020.

Exemple

Use of variables L

In program %1, block N100 is not prepared and executed until block N90 is completed.

In program %2, block N100 is prepared before execution of N80 and there is no stop at the end of block N90.

```
%1
N.. ...
N.. ...
N90 X.. Z..
N100 L110 = 1 + L110
N110 XL110
N120 L120 = L110 + L10
N130 ZL120
N..
N..
```

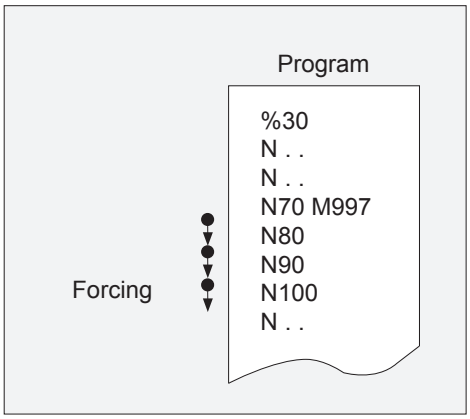
```
%2
N..
N80 M999
N90 X.. Z..
N100 L110 = 1 + L110
N110 XL110
N120 L120 = L110 + L10
N130 ZL120
N140 M998
N..
```

5.19.10 Forced Block consecutive execution ► M997

M997	Forced block consecutive execution.

When this function is active continuous execution of program blocks is forced either in Single mode or MDI.

This is generally use to perform operation that should be made in sequence (e.g. Tool changer).



Syntax

N.. **M997**

M997 Forced block continuation.

Properties of the Function

Function M997 is a decoded modal «pre» function.

Cancellation

Function M997 is cancelled by functions M998, M999 and M02.

Notes

Use of function M997 in single step (SINGLE) mode

If the operator starts program execution in SINGLE mode, reading M997 during the program causes sequencing of the following blocks as though the system were in continuous mode.

Use of function M997 with a subroutine call by M function requested in manual data input (MDI) mode

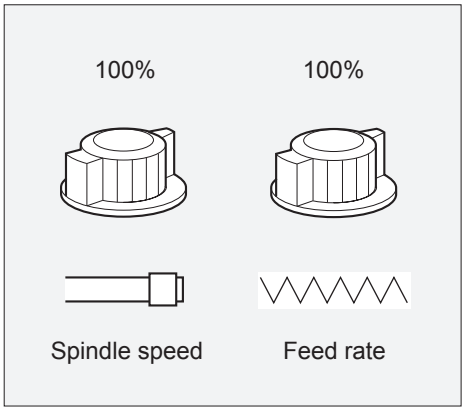
Programming this function causes execution of the subroutine like in continuous mode.

5.19.11 Potentiometer Inhibit ► M49, M48

M49	Spindle speed and feed rate potentiometer inhibit.

During execution of the program, the operator cannot override the spindle speed and feed rate. The potentiometers are forced to 100 percent.

M48	Spindle speed and feed rate potentiometer enable.



Syntax

N.. **M49/M48**

M49	Spindle speed and feed rate potentiometer inhibit.
M48	Spindle speed and feed rate potentiometer enable.

Properties of the Functions

Function M48 is a modal «post» function. It is the default function.

Function M49 is a modal «pre» function.

Cancellation

Functions M48 and M49 cancel one another.

Notes

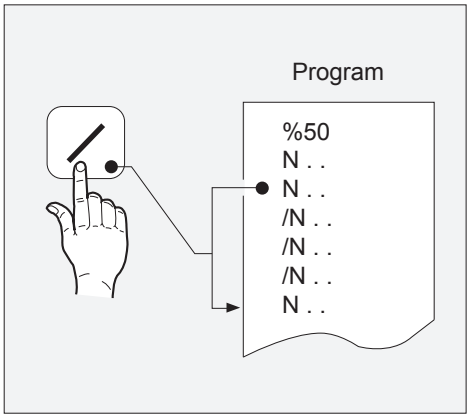
Programming M49 causes:

- forcing of a feed rate of 100 percent (with M48, feed rate override possible from 0 to 120% of the value programmed with F),
- forcing of a spindle speed of 100 percent (with M48, spindle speed override possible from 50 to 100% of the value programmed with S).

5.19.12 Block Skip ► / N

/ Block skip.

A block preceded by a slash «/» is ignored when block skip is enabled by the operator.



Syntax

/ N.. (Contents unimportant)	
/	Block skip (/).
N..	Block number.

Cancellation

By inhibiting block skip.

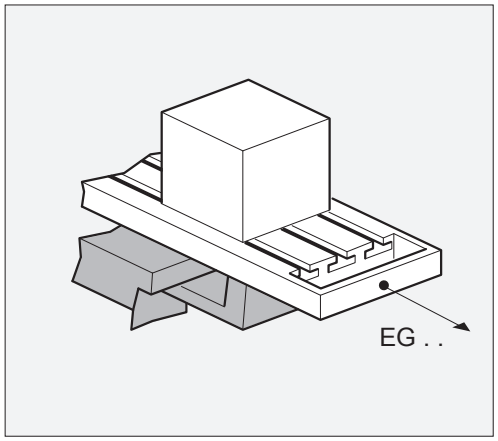
Notes

Block skip «/» is active when confirmed by the operator (the «/» indicator is displayed in the status window). Bloc skip validation cannot be changed during program execution.

5.19.13 Acceleration Reduction ► EG

EG..	Programmed acceleration reduction.

This function followed by a value sets the maximum acceleration tolerated on the programmed movements. This allows the stresses due to movement of heavy loads to be limited.



Syntax

N.. **EG**..

EG..	Programmed acceleration reduction. A positive integer giving the percentage (between 1 and 100%) of the values set by the machine builder.

Property of the Function

Function EG.. is modal.

Function EG is forced to 100 percent at power on.

Cancellation

Function EG is cancelled by:

- programming a new value (EG..),
- the end of program function (M02),
- a reset.

Notes

On the information (INFO) page:

- EG1 to EG99 is displayable,
- EG100 is not displayable.

During programmed interpolated movements, the reduced acceleration is always taken into account by the interpolators except in the case of Cycle stop or activation of the feed safety device.

Acceleration reduction is ignored in the following modes or functions:

- axis jog (JOG),
- homing (HOME),
- action after a program stop (M12).

Example

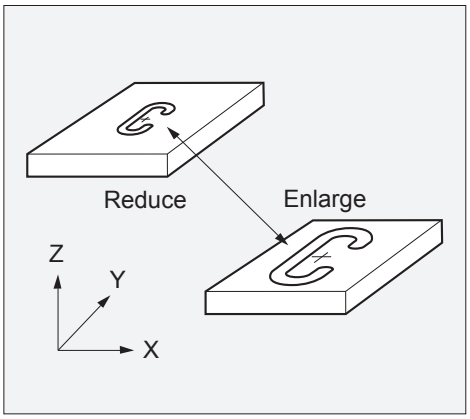
```
N.. ...
N40 EG50 G00 X..          Acceleration reduced to 50% on X
N..
```

5.19.14 Scaling Factor ► G74, G73

G74	Scaling factor enable.

This function allows execution of scaled parts from a single programmed part or template. The scale ratio can be entered from the keyboard or programmed.

G73	Scaling factor cancel.



Syntax

N.. [G40] **G74/G73**

G74	Scaling factor enable. The ratio declared can be between 1 and 9999 to 1000 (0.001 to 9.999) and it must be an integer.
G73	Scaling factor cancel.

Properties of the Functions

Functions G73 and G74 are modal.

Function G73 is the default function.

Cancellation

Functions G73 and G74 cancel one another.

Function G74 is cancelled by the end of program function (M02).

Notes

Scaling is around the program origin (OP).

The ratio can be entered from the keyboard or programmed by external parameter E69000.

Functions G73 and G74 must be programmed:
- with the system in G40 mode (radius offset cancel),
- in a block that does not contain any circular interpolation,
- outside a sequence of incompletely defined PGP (Profile geometry Programming) blocks.

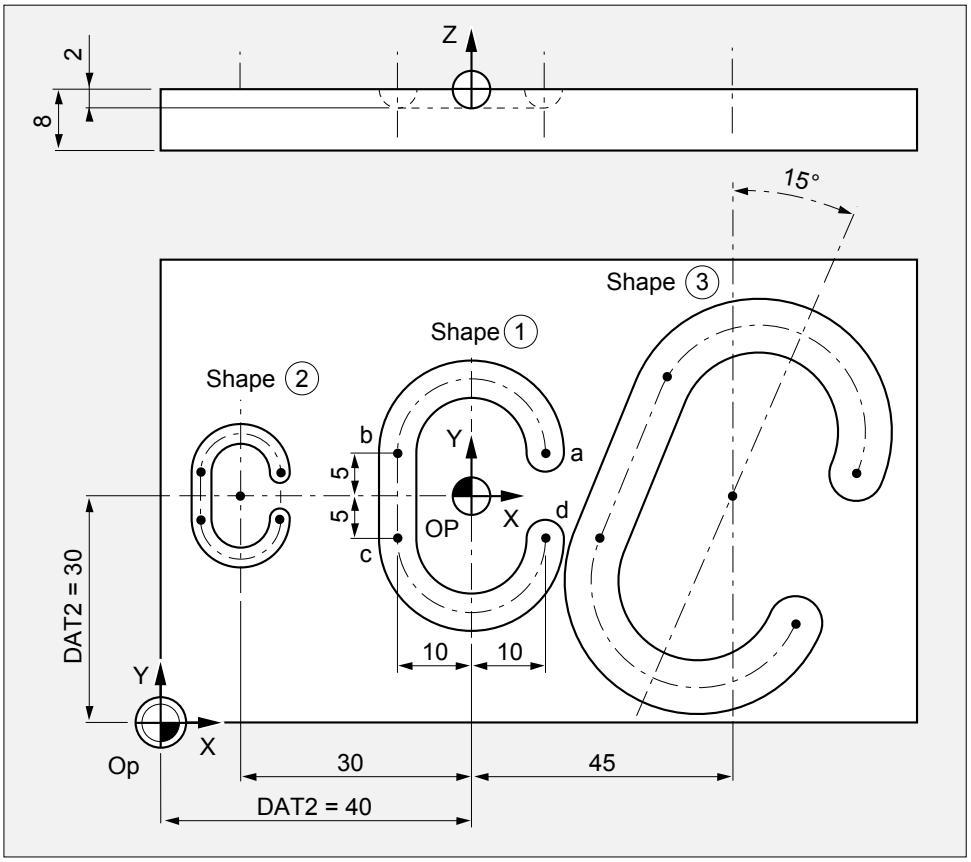
The ratio affects:
- the values programmed for the primary and secondary axes (X, Y, Z, U, V, W),
- the program origin offset (G59).

The ratio does not affect:
- the values programmed for the rotary axes (A, B, C),
- the part origin DAT1,
- the shift between the part origin and the program origin (DAT2),
- table off-centering (DAT3),
- tool dimensions (L, R, @),
- programming with reference to the measurement origin (G52),
- the clearance dimension for canned cycles G81 to G89.

Please refer to the Flexium HMI manual M00017 for operation of entering the scaling factor on the keyboard.

Example

Execution of shapes using scaling factors



- Shape 1: Scale 1 from the program zero.
- Shape 2: Scale 0.5 with origin offset (G59) on X-30.
- Shape 3: Scale 1.5 with origin offset (G59) on X45 and angular offset (ED) of 15 degrees.

As the origin offset is affected by the scaling factor, values assigned to G59 must be proportional to the programmed scaling factor.

Main program example

```
%300
N10 G00 G52 Z0
N20 T01 D01 M06 (BALL END CUTTER, DIAMETER=8)
N30 S2000 M03 M40
N40 G59 X0 Y0                          Zero origin offset
N50 G74 E69000 = 1000                  Scaling factor 1
N60 G77 H3                             Subroutine call for shape 1
N70 G59 X-60                           Origin offset
N80 G74 E69000 = 500                   Scaling factor 0.5
N90 G77 H3                             Subroutine call for shape 2
N100 G59 X30 ED-15                     Origin offset
N110 G74 E69000 = 1500                 Scaling factor 1.5
N120 G77 H3                            Subroutine call for shape 3
N130 G00 G52 Z0 M05 M09
N140 M02
```

Subroutine example

```
%3
$0 SHAPE C
N10 X10 Y5 Z3                          Point a
N20 G01 Z-2 F100 M08                   Approach on Z
N30 G03 X-10 Y5 R10 F300               Point b
N40 G01 Y-5                            Point c
N50 G03 X10 Y-5 R10                    Point d
N60 G73 G00 Z3                         Scaling factor cancel
```

Example

Programming the scaling factor by external parameter E69000

External parameter E69000 is a read/write parameter. The value programmed must be an integer.

```
N.. ...
N40 E69000 = 250                       Ratio 250/1000, i.e. reduction to 0.25
N50 G74 G00 X.. Z..
N..
N200 G73                               Scaling factor cancel
N..
```

The scaling factor can be tested by a conditional branch in the program.

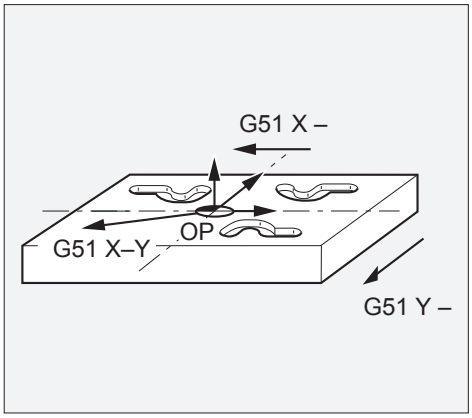
```
N.. ...
N.. G79 E69000 = 300 N210              If the ratio is equal to 300, jump to block N210
N..
N..
N210
N..
```

5.19.15 Mirror function ► G51

G51	Mirror function.

This function allows symmetrical machining of program sections defining a quarter or half of the part.

The mirror is enabled or cancelled according to the axis and algebraic sign arguments programmed with the function.



Syntax

N.. **G51** X- Y- Z- A- B- C-

G51	Mirror function.
X- Y- Z- A- B- C-	The minus sign (-) enables the mirror function on the X, Y, Z axes or A- B- C-.

Properties of the Function

Function G51 is nonmodal. The axis arguments (X, Y, Z, A, B, C) associated with the function are modal.

Cancellation

Mirroring on the programmed axes is cancelled by:
- function G51 followed by one or more arguments X+, Y+, Z+, A+, B+ or C+ cancelling the former state G51,
- the end of program function (M02),
- a reset.

Notes

When G51 is programmed:
- it must be programmed with at least one argument (axis and sign),
- it must be programmed alone with its arguments in the block,
- several axes can be enabled or inhibited in the same block,
- the axes to which the mirror function applies are displayable on the information page (INFO),
- if a mirror function is applied to a carried axis, it is also automatically applied to the carrier axis.

The mirror function affects:
- the sign of the programmed axis X, Y, Z, A, B or C which is reversed. The sign change is carried out with respect to the program origin defined by DAT1 and DAT2,
- the origin offsets programmed (G59),
- tool radius offset (G41, G42),
- the direction of rotation with circular interpolation (G02, G03).

The mirror function does not apply to:
- the part origin DAT1,
- the shift between the part origin and the program origin (DAT2),
- table off-centering (DAT3),
- the tool dimensions (L, R, @),
- programming with respect to the measurement origin (G52).

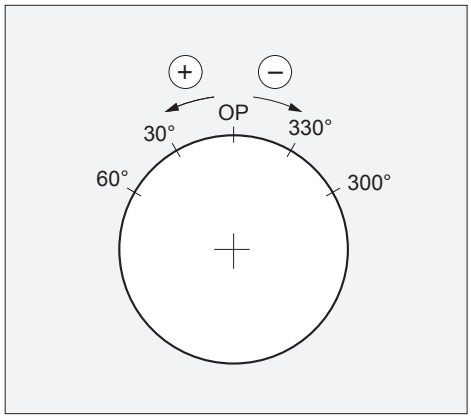
When the mirror function is enabled on the axis corresponding to the tool orientation axis, a new orientation must be programmed (G16 ...).

5.19.15.1 Influence of the mirror function on the A, B or C rotary axes

The function reverses the direction of rotation of the rotary axes and takes the complement to 360 degrees.

If B30 is programmed: the axis moves to 330 degrees in the negative direction.

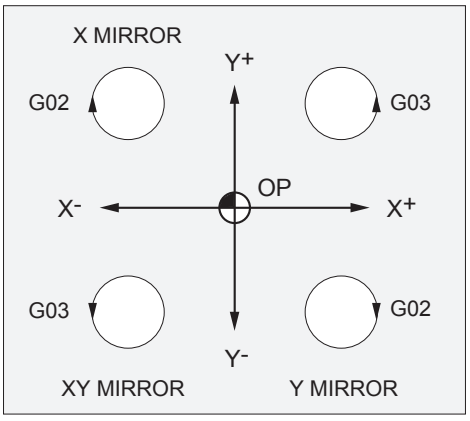
If B-300 is programmed: the axis moves to 60 degrees in the positive direction.



When the mirror function is applied to a rotary axis, it should be checked by a test before machining that the axis rotates in the required direction.

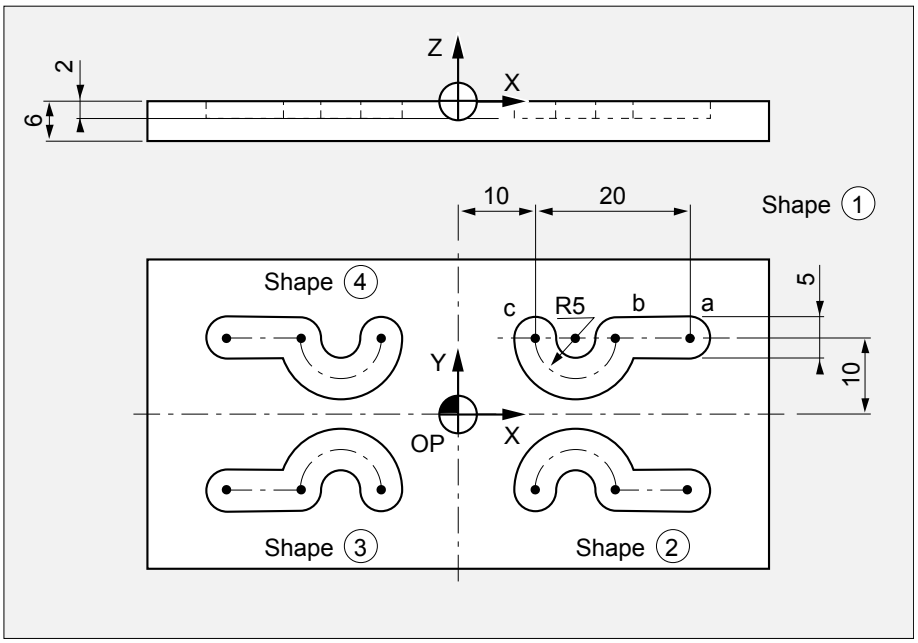
5.19.15.2 Influence of the mirror function on circular interpolations

The function reverses the clockwise (G02) and counterclockwise (G03) rotations.



Example

Execution of a shape using mirror function G51 in the XY plane




```
%30
N10 G00 G52 Z0
N20 T05 D05 M06 (CUTTER DIAMETER=5)
N30 S1500 M40 M03
N40 G00 X30 Y10 Z2      Point a, approach on XYZ
N50 G01 Z-2 F50 M08     Penetration on Z
N60 X20                 Point b
N70 G02 X10 Y10 R5      Point c
N80 G00 Z2              Retraction on Z
N90 G51 Y-              Mirror on Y
N100 G77 N40 N80        Execution of shape 2
N110 G51 X- Y-          Mirror on X and Y
N120 G77 N40 N80        Execution of shape 3
N130 G51 X- Y+          Mirror on X, cancellation of mirror on Y
N140 G77 N40 N80        Execution of shape 4
N150 G51 X+ Y+          Cancellation of mirror on X and Y
N160 G00 G52 Z0 M05 M09
N170 M02
```

5.19.16 Processing of Blocks and Programmed G and M Functions ► G999, G998, G997

G999	Suspension of execution and forcing of block concatenation.

The blocks programmed after G999 are concatenated. The movements are no longer made on the axes and functions M, S and T are no longer processed.

These functions are used for creating custom cycles. They are not used in standard part programming and are explained more in details in the Extended Programming manual.

G998	Enabling of execution of the blocks and part of the functions processed in state G999.

The blocks including movements and functions processed in state G999 are enabled and executed with the exception of certain functions that are only stored (decoded M «post» functions, dwells and synchronisation functions for multiple channels).

G997	Enabling and execution of all the functions stored in state G999.

All the functions programmed (without exception) are enabled and executed, including those processed in state G999.

Syntax

N.. **G999** / **G998** / **G997**

G999	Suspends execution and forces block concatenation.
G998	Enables execution of the blocks and part of the functions processed in state G999.
G997	Enables and executes all the functions stored in state G999.

Properties of the Functions

Functions G999, G998 and G997 are modal.

Cancellation

Functions G999, G998 and G997 cancel one another.

Notes

Notes on Function G999

After execution has been suspended and the blocks have been concatenated by G999, analysis of the following blocks continues and the values encountered in these blocks of the part program are stored in special symbolic variables.

These variables can be used to perform computations or read programmed values, but the axes are not moved and the M, T and S functions are not processed.

A subroutine call by function Gxxx (see Flexium Extended Programming manual M00020) implicitly sets function G999. This function then has to be cancelled by including functions G998 and/or G997 in the subroutine.

During the return to the part program, state G999 is systematically reset as long as the subroutine call function (Gxxx) remains present and active (no G80).

Notes on function G998

Function G998 enables and executes:
- movements on the axes,
- auxiliary functions such as spindle speed (S), tool number (T), decoded «pre» M functions.

Function G998 stores the following functions but does not enable them:
- decoded «post» M functions (e.g. M05, M09, etc.),
- dwells (G04 F..),
- axis multichannel synchronisation functions (G78 ...).

Example

Sequencing of functions G999, G998 and G997

```
%55
N.. ...
N.. G999 S2500 M03
N.. G00 X100 Y50 G04 F5
N.. G01 Z10
N.. ...
N.. M05
N.. G998


N.. ...
N.. ...
N.. X200 T02 M06
N.. ...
N.. G997
```

Blocks concatenated by G999

Enabling of S2500 M03 («pre» M function) and movements on the axes to X100, Y50 and Z10 in G01 mode at F5

Movement to X200 and enabling of T02 M06

Enabling of G04 F4 (dwell) and M05 («post» M function)

5.19.17 Tridimentional curve smoothing ► G104

G104	Tridimentional curve smoothing.

This function executes tridimentional curves of any shape. It automatically performs polynomial interpolation between a sequence of points defined by their X, Y and Z coordinates. Curve smoothing must be programmed by a minimum of three blocks.

General Syntax

N.. X.. Y.. Z..

N.. [G01] **G104** X.. Y.. Z.. [F..]

[Intermediate points]

N.. G80 X.. Y.. Z..

X.. Y.. Z..	Coordinates of the curve start point.
G104 X.. Y.. Z..	Curve smoothing function and coordinates of the second point on the curve.
Intermediate points	Sequence of blocks each containing the coordinates of the intermediate points on the curve (optional).
G80 X.. Y.. Z..	Cancellation function G80 followed by the curve end point coordinates.

Property of the Function

Function G104 is modal.

Cancellation

Function G104 is cancelled by function G80.

Notes

The block containing the curve smoothing start point must always be programmed before calling function G104. The block containing the end point must always include function G80.

As the function is interpolated directly by the system without prior computations, there is no limit to the number of points in the curve and no stop at the end.

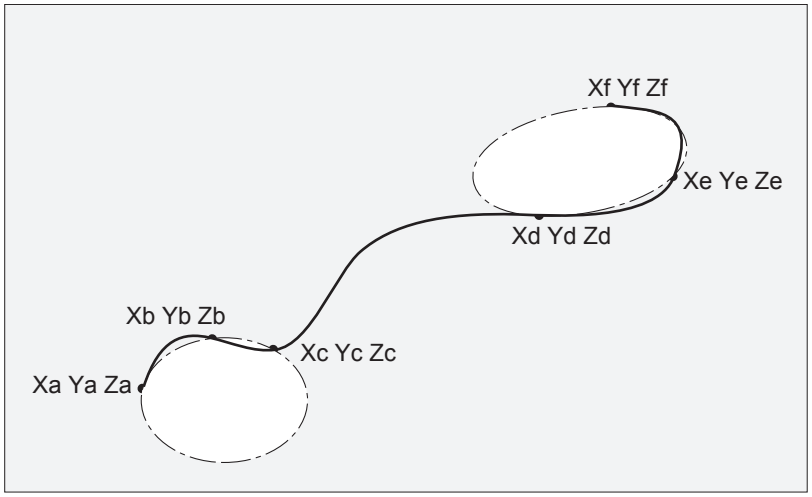
The blocks defining curve smoothing may contain:
- Miscellaneous M functions,
- Program variables L,
- External parameters E.

Curve smoothing can be executed in all the CNC Modes: AUTO, SINGLE, etc. The SEARCH mode can be used for partial execution of a curve.

The location of each point on the curve can be captured empirically by "Teach-in".
(see Flexium HMI Operator manual M00017).

Example

Programming of curve smoothing to execute polynomial interpolation between points a and f




```
%123
N.. ...
N.. ...
N100 Xa Ya Za
N110 G01 G104 Xb Yb Zb F200
N120 Xc Yc Zc
N130 Xd Yd Zd
N140 Xe Ye Ze
N150 G80 Xf Yf Zf
N.. ...
M02
```

Curve start point
Smoothing function and second point

Intermediate points

Cancellation and end point

The tangents to the end points (points a and f) are the tangents to the circles going through the first three points (a-b-c) and the last three points (d-e-f) of the curve.

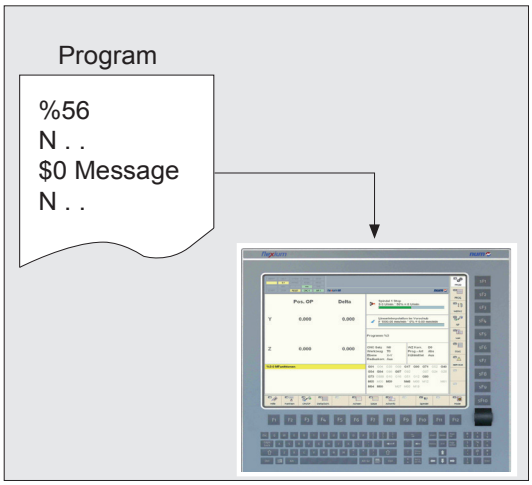
5.20 Message Transmission ► $0

The $ character, optionally, followed by a zero sends a message from a part program to the display.

5.20.1 Message Transmission to the Display

$0	Message transmission to the display.

$0 sends the message to the system display (information message on the part program being executed).



Syntax

$0 MESSAGE or $+ Message complement

$0	Message sent to the display.
$+	The «+» sign extends the previous message.
MESSAGE	Message containing a maximum of 39 (alphanumeric) characters.

Cancellation

- $0 (without a message).
- End of program (M02).
- Reset.

Notes

The message sent by $0 can be read on the Messages area of HMI.

It should be noted that:

- If a message is too long, only the first 39 characters are displayed.
- Only one message can be sent at a time.
- Sending a new message overrides the previous one.
- The 0 after $ is optional, but it is recommended.

Extensions to the use of the $ character

The $ character can be used in parametric programming to:

- Display a message and wait for an answer from the operator.
- Display a message with a parametric value.

Examples

The message beginning in block N190 and including an extension is displayed in the following format:

«PROGRAM HALT. REMOVE CHIPS»

```
%30
$0 GROOVE ROUGHING FINISHING            Message
N10 ...
N..
N190 G00 Z300 $0 PROGRAM HALT.          Message
$+ REMOVE CHIPS                         Message extension
N200 M00
N.. $0                                  Message cancellation
N..
```

Exemple

Use of $0 as flashing message

```
N..
N240 $0 * * * PREPARE PART INSPECTION * * *
N250 G04 F0.6
N260 $0 * * * <<<<<<<<<====>>>>>>>>> * * *
N270 G04 F0.3
N280 G77 N240 N270 S4
N290 M00 $0 * * * OPEN DOOR * * *
N300 $0
N..
```

5.21 Message Display with request for an Operator Response ► L../E..,$

The «$» character can be used as variable in a parametric expression. It causes the program to wait for an operator entry.

The block containing the parametric expression can be preceded by a message indicating the type of data to be entered.

Syntax

L../E.. = Expression depending on $

L../E..	Program variable or external parameter.
$	Variable whose value must be entered by the operator.

Cancellation

- end of program (M02),
- reset.

Notes

Upon reading the «$» character associated with a variable (possibly accompanied by a message with a maximum of 39 characters), the system stops the program and waits for the operator to type an entry.

The characters typed are echoed on the upper right window of the HMI.

The entry can be numerical characters or one alphabetic character. Each type of entry corresponds to a specific use.

Numerical entry

The number of digits entered must not exceed eight plus sign and decimal point.

The limit values are as follows:
- Integer: ≤ 99999999 or ≤ -99999999
- Positive decimal: ≤ 0.0000001 or ≤ 9999999.9
- Negative decimal: ≤ -9999999.9 or ≤ -0.0000001.

The operator must confirm the entry by pressing the Enter key on the keyboard (otherwise the system does not proceed to the next block).

Before pressing Enter, the characters entered can be deleted by the Delete key.

Alphabetic entry

Only the first character entered is analysed.

Any capital letter from A to Z is accepted. Each letter returns a value:

- A returns 1
- B returns 2

and so forth to Z = 26.

It should be noted that in a test program, each numerical value from 1 to 26 corresponds to alphabetic characters A to Z.

Example

Message Display

```
N.. ...
N50 E80000=50
$0 ENTER THE VALUE OF X =          Display of the message for the operator
N60 L0=E80000 + $                  Wait for entry of the value
N..
```

5.22 Display of Messages with Parametric Value ► $

The «$» character can be used as the first term of a parametric expression whose value is displayed after being calculated by the system.

The block containing the parametric expression can be preceded by a message indicating the name of the variable displayed.

Syntax

$ = expression

$	Variable whose value is displayed.

Cancellation

- end of program (M02),
- reset.

Notes

An expression «$ =...» in a block after the branch function (G79) cannot be used as a branch condition.

Example

```
N..
L0=0                          Initialisation of variable L0
N90 L0=L0+1                   Increment of L0
$0 EXECUTION OF SUBROUTINE    Display of the first part of the message
$= L0                         Display of the value following the previous message
G77 HL0                       Call to subroutine % (L0)
G79 L0<5 N90                  Condition: if L0 < 5, branch to block N90, else continue
N..
```

5.23 Transfer of the Current Values of L Variables and E Parameters into the Part Program ► G76

G76	Transfer of the current values of L variables and E parameters into the specified program or program section.

This function is used to update the contents of a file called by addresses H.. and/or N.. N..

The file of L variables and E parameters is updated with the new contents of the corresponding active data.

Syntax

N.. **G76** [H..] [N.. N..]

G76	Transfer of the current values of L variables and E parameters into the program specified.
H..	Specification of the program into which the values are transferred.
N.. N..	Specification of the program section into which the values are transferred.

Notes

The parameters to be transferred must be located at the beginning of blocks: L variables and E parameters located after another function in a block are ignored.

The specification of an L variable or E parameter must be followed by the sign «=» and at least ten characters (spaces, algebraic sign, numbers, decimal point) designed to be replaced by a new value.

Failure to comply with this rule returns error message 97.

The file modified with G76 is stored in NC volatile memory. It is saved only during a proper shut down of the system. Make sure the file has been properly saved before using the results.

Examples

Transfer of the values of L and E into the current program

In the example below, only parameters L101, E80001, L4 and L6 are modified. The other parameters are ignored.

```
N..
N40 G76 N100 N120
N..
N100 ..
L101=..... E80001=.....
L4=..... G04 E52002=.....        E52002 is not modified as it is located after G04
L6=.....
G01 X100 L3=.....                L3 is not modified
N120
N..
```

Transfer of the values of L and E into a subroutine

```
%125
N10 G77 H200 N50 N80             Execution of blocks N50 to N80 of %200
N..
N..                              Execution of the program after modification of the
                                 parameters by %200
N..
N600 G76 H200 N50 N80            Update of the file
N610 M02


%200
N10 ...
N..
N50
L1=..... E52002=.....
E80004=.....
N80
N..
```

5.24 Special Programming for Multiple Channels

5.24.1 Program Declaration

A machining program for multiple channels is an association of the programs for each axis channel.

The program names have a common radical followed by an index specifying the channel, e.g.:

- %61.1 Program number for channel 1
- %61.2 Program number for channel 2

Program number format: %05.1 (indices .1 to .8).

The programs for each of the channels must be loaded in the NC memory. Otherwise, the machining program cannot be started (unless one or more channels are inhibited).

5.24.2 Programming Notes

It is essential not to use the program start symbol «%» in comments in the program body.

Programmable Axes

The NC can control 32 axes in a maximum of 8 channels.

Each channel can include up to 9 axes declared by the machine manufacturer.

Example: Machine with 9 axes in 3 channels

- Channel 1: axes X Y Z A
- Channel 2: axes U V W B
- Channel 3: axis C

Axes in different channels can have the same name. In the above example, axes U V and W of channel 2 could have been named X, Y and Z.

For part and program set-up it is possible to selectively inhibit a particular channel. Please refer to the manufacturer's manual for details.

T Function

The machine manufacturer has been given two registers for the T address.

The same tool number T.. can be used for different tools one in the program with index .1 and in one other program, e.g.:

```
%10.1                          %10.2
N.. ...                        N.. ...
N.. T05 M06 (DRILL)            N.. T05 M06 (CUTTER)
N..                            N..
```

Different tool numbers must be used for the programs with indexes .2 to .8.

Tool Corrections

The tool correction tables are common to all the programs.

Program Variables

L variables can be used in each program.

L0 to L19, L100 to L199 and L900 to L959: these variables can be used for each axis channel (no interaction between programs).

External Parameters

External parameters E are common to all the programs (except for parameters E50000, E51000, E6x000 and E7x000 which relate to their individual axis channel).

Subroutine branches for Multiple Channels

Subroutines which are not part of the machining program must always include the index of the calling channel. For instance: %xxx.i (i = channel number).

Branch to Automatic Homing Subroutine

Homing can be carried out automatically on each of the axes of the channel by running subroutine %9990.i (i = channel number).

Requirements for running subroutine %9990.i

Action on the cycle start button in homing mode runs subroutine %9990.i for the channel if no other program is being executed.

The automatic homing program is run simultaneously on all the existing CNC channels.

Subroutine %9990.i can also be called as a subroutine from another program while preserving the property of being able to move the machine axes before being homed.

This possibility can be used for homing an Auxiliary channel. If such a program can be used in homing mode or as a subroutine, it can also end as follows:

```
IF [.IRH(1)] = 9990.i THEM M02          If it is not from a subroutine, then set M02
ENDI
```

For information on the above programming, refer to the Flexium Extended Programming manual M00020.

Subroutine Call by a Reset

On a reset, the NC channels can execute subroutines with numbers %11000.i (i = channel number).

Aside from the fact that the subroutine numbers include the index (i) of each NC channel, the call is the same as the subroutine call by reset for a single channel.

Subroutine Call by the PLC

The automatic control function can call and execute a subroutine with number %9999.i (i = channel number).

The call is the same as a subroutine call with a single channel except for the CNC channel number (i).

Subroutine Call by M Function

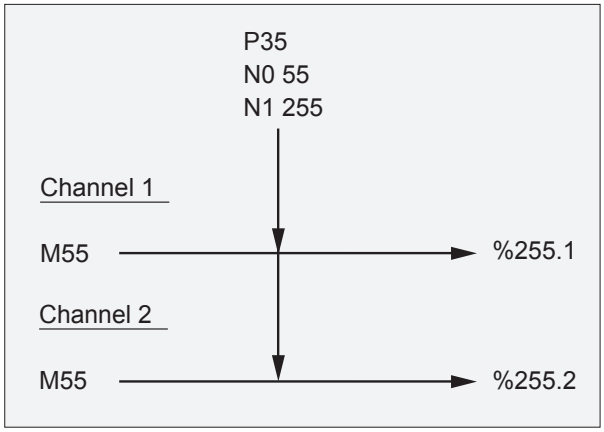
An M.. function can call and execute a subroutine with number %xxx.i (i = channel number).

The call is the same as a subroutine call with a single channel except for the CNC channel number (i). The association between M function and called subroutine is done into parameter P35.

The assocuiation between the M function and the program to execute is defined by the machine builder in the parameter P35.
Please refer to the machine user's manual for the actual correspondance.

Example

With two channels (1 and 2), call of two subroutines %255.1 and %255.2 with different contents by function M55.



5.24.3 Programming the Spindles

The spindle characteristics are defined by the machine builder.

A spindle can be:
- controlled by a channel,
- independent.

Spindle control by a channel

A spindle can only be controlled by a single channel at a time, and a channel can only control one spindle at a time.

In a channel, selection of another spindle by one of functions M62 to M65 is ignored unless the spindle is available, ie not already controlled by any other channel (otherwise, the system returns error message 38).

Independent spindle

An independent spindle is a spindle which is not being controlled by any channel or which has been released by the channel that was controlling it.

A spindle is released from a channel by function M61.

Only an independent spindle can be selected by a channel for control by that channel.

Spindles not assigned to a channel are independent.

Reminder

On a machine equipped with a milling spindle and a turning spindle with indexing performed by the NC:
- the milling spindle is indexed by function EC..,
- the turning spindle is indexed by positioning on the C axis.

5.24.4 Releasing the Current Spindle in the Channel ► M61

M61	Release of current spindle in the channel.

This function releases the current spindle in the channel to allow it to be controlled by another channel.

Syntax

N.. **M61**

M61	Release of current spindle in the channel.

Properties of the Function

Function M61 is a modal «post» function.

Cancellation

Function M61 is cancelled by functions M62 to M65.

Notes

After release of a spindle by function M61, the spindle can be controlled by one of functions M62 to M65 in the program of another channel.

When the spindle to be controlled has not been released, the system returns error message 38.

Example

Program for channel 1	Program for channel 2	
%40.1	%40.2	
N.. ...	N.. ...	
N..	N.. G97 S500 M03 M65 M40	
N..	N..	
N..	N.. M61	Release of spindle 2
N..	N..	
N.. G97 S1000 M03 M41 M65	N..	
N..	N..	

5.24.5 Channel synchronisation with markers ► G78

G78	Channel synchronisation with markers

This function identifies and manages the synchronisation steps in the execution of each program.

Syntax

N.. **G78** Q.. Pj.i

G78	Channel synchronisation.
Q..	Declaration of a marker in the current channel.
Pj.i	Wait for a marker from another channel. (optional)
	Argument P is defined by two digits separated a decimal point:
	- j is the marker number tested, - i is the index of the channel whose marker is tested.

The P argument can be repeated if necessary for other channels.

Example: `G78 Q5 P3.2 P5.3 P2.4`

Defines marker 5 for the current channel, waits for marker 3 passed on channel 2, marker 5 passed on channel 3 and marker 2 on channel 4.

Property of the Function

Function G78 is nonmodal.

Cancellation

Function G78 is cancelled at the end of the block.

Argument Q associated with the function has a default value of zero (Q0).

During the program, the markers can be reset by programming G78 Q0.

Notes

Function G78 can be followed by several arguments, but at least one is mandatory.

The declaration of a marker and the program continuation conditions can be programmed in the same block, e.g.:

```
N.. G78 Q3 P5.2 P6.3
```

The channel selected and the associated step transition conditions can be enabled selectively by the automatic control function.

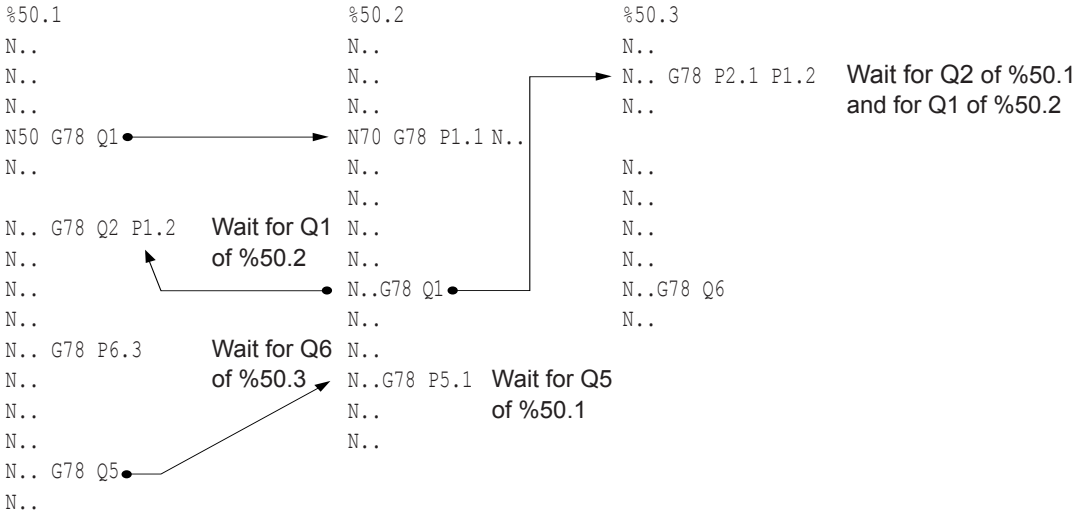
The markers included in a program must be numbered in increasing order, but not necessarily sequentially, e.g.:

```
N.. ...
N90 G78 Q1 ...
N..
N200 G78 Q6 ...
N..
```

Channel markers are set as their programs reach them.

Example

Program synchronisation with markers



Sequence N70 of program %50.2 is crossed if program %50.1 reaches or has passed sequence N50.

Program %50.3 does not start until program %50.1 reaches marker Q2 and program %50.2 reaches or passes marker Q1.

If channel 3 is inhibited by the PLC program, the steps related to program %50.3 are ignored in programs %50.1 and %50.2 (wait G78 P6.3 of %50.1 is ignored).

5.24.5.1 Programming a Meeting Point

A meeting point can be programmed by function G78 or initiated by programming M functions.

5.24.5.2 Programming by function G78

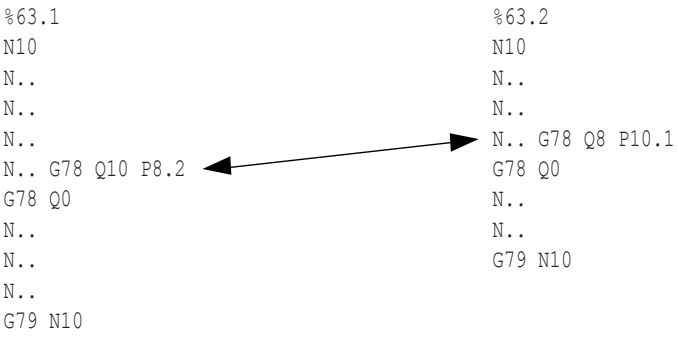
When a meeting point has been programmed, the programs do not continue until the other programs have reached their respective markers.

When the meeting point has been reached by all the channels, the markers can be reset by programming G78 Q0.

Before the markers are reset by G78 Q0, the previous block must mandatorily be a meeting point, i.e. a complete synchronisation on all the programs.

Example:

Programming a meeting point on two channels (with complete synchronisation).

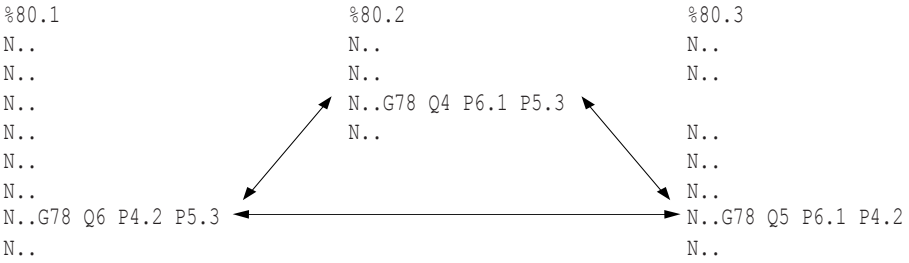


The end of program command (M02) cancels the conditions of that channel and is equivalent to inhibiting the program.

If all the programs are on wait, the system returns error message 33 (synchronisation error). If this error is detected in test mode (TEST) or sequence number search mode (SEARCH), it is necessary to carry out a reset and to correct the program.

Example

Programming a meeting point with G78 on 3 channels



The three programs continue together when markers Q6, Q4 and Q5 are reached.

5.24.5.3 Programming with M functions

The following functions are used as meeting points:

- programmed feed stop (M12),
- program stop (M00),
- optional stop (M01).

When M00 or M01 (enabled) are programmed in only one of the programs, the movements of the axes in the channel are stopped and the system waits for the end of the other programs (M02) before executing M00. When the cycle restarts, only this program continues.

It should also be noted that functions M00 and M01 (even not enabled) reset the markers.

When M01 is programmed in all the programs but was not enabled, each of the M01 sequences is a meeting point, but the cycle continues automatically when all the programs are on their respective M01 commands.

5.25 Special Programming of Auxiliary channels

5.25.1 Program Declaration and Storage

The programs executed by the auxiliary channels are identified by the radical 9998 followed by an index defining the channel, e.g.:

- %9998.2 Program executed by auxiliary channel 2.
- %9998.3 Program executed by auxiliary channel 3.

Program number format: %04.1 (indexes 2 to 8).

The programs and subroutines must be stored in a CNC memory zone different of the User Zone (Customer, OEM or NUM zone).

When the main program is stored after creation or replaced after editing, the channel must be reset so that its presence is taken into account.

Structure of a auxiliary channel program

When different functions should be processed by an auxiliary channel, the PLC must specify the function called, which can be done by external parameter E40000, for instance.
At the start of the program:

- an M function is used as report
- external parameter E40000 contains the sequence number of each function requested (Na, Nb, etc.).

Example

Program %9998.2 (auxiliary channel 2) below contains several functions.
The second program block defines a jump to the sequence number contained in parameter E40000.

```
%9998.2
N1 M..                          Wait for start by AckMFunction
G79 NE40000                     Jump to the sequence defined

Na                              Process the first function
..
..
..
G79 N1                          Jump to the sequence

Nb                              Process the second function
..
..
..
G79 N1                          Jump to the sequence
..
```

5.25.2 Programming of the Auxiliary channels

The auxiliary channel are programmed in the same way as the NC channel but there are restrictions concerning the use of certain functions.

Restrictions

The following functions are unavailable:
- optional stop (M01),
- programmed feed stop (M12).

5.25.2.1 Emergency Retraction on an Auxiliary Channel

Emergency retraction (function G75...) on an auxiliary channel allows a particular sequence to be executed in automatic mode at the request of the PLC when the same channel is in a cycle.
It should be noted that:
- No particular condition is required as to the current mode on the other channels
- The emergency retraction request on an auxiliary channel only affects that channel.

Activation of emergency retraction

Function G75 N.. defines a branch to a sequence N.. when emergency retraction is activated.
Activation is cancelled by declaring a new address G75 N.. or G75 N00.
The emergency retraction is activated from the PLC either automatically or on a manual action.
For further information please refer to the Machine Operation Manual.

It should be noted that:
- If the channel is executing a cycle (automatic mode «AUTO» or single step mode «SINGLE»), activation halts the program being executed. The halt is followed by a branch to sequence N.. programmed after G75 followed by execution of the retraction program until one of functions M00 or M02 is encountered.
- Signal Flexium_NCK.RCNC.Channel[0].EmergencyRetract remains set throughout the duration of the emergency retraction program. This signal is reset by encountering function M00 or M02 which ends the emergency retraction program.

If emergency retraction is activated but no G75 N.. was programmed, this activation stops the axes of the channel and resets the channel (RAZ).

5.25.2.2 Exchanging Axes between Channels

The channels can exchange axes via the part program. The axes are exchanged using external parameters E7x005 (x = axis number).

After a reset, the channel (PLC or NC) releases the axes that were not assigned to it by machine parameter and assigns them to the priority channel providing the priority channel is:
- in the end of program state (M02), or
- with no program being executed for an auxiliary channel.

Similarly, during a reset concerning it, a channel can recover the axes released by other channels that are assigned to it by machine parameter:
- after a reset,
- by programming external parameter E7x005.

It should be noted that if a channel is not in state M02 when its axes are released, it cannot recover them until the next reset concerning it, i.e.:
- an NC reset for an NC channel,
- a reset on the channel for an auxiliary channel.

When the cycle starts, no checks are made on assignment of the axes to the channel. All the axes assigned to the channel after the last reset were assigned by the machine builder, but certain axes assigned to it after the first initialisation may not be reassigned if they have already been assigned to another channel during a cycle (it is therefore up to the programmer where required to ensure the configuration of the axes in the channel in order to authorize start of the cycle).

5.25.2.3 Exchanging Spindles Between Channels

When the NC is initialised, if no spindle has been assigned to a particular channel, the spindles are assigned to the channels that have the same suffix, e.g.:
- Spindle 1 to channel 1,
- Spindle 2 to channel 2, etc.

An auxiliary channel A can be assigned a spindle if the spindle was first released by its owner channel B.

When channel B, the initial owner, is reset, the spindle is reassigned to channel B (even if channel A is executing a cycle).

5.26 Special Features of Mixed Machines (MX)

Machines combining standard milling functions with turning functions are called mixed machines.

Such mixed machines involve:
- Special features related to the machine axes,
- Special programming of certain ISO functions.

Below, the descriptions are limited to the following types of machines:
- borers,
- mixed machines.

5.26.1 Features Related to the Machine Axes

Programmable axes on MX:
- Primary axes X, Y and Z,
- Secondary axes U, V and W,
- Rotary axes A, B and C.

Axis Declaration

Boring type machines

For such machines, radial axis U must be declared independent.

Borers or mixed machines

On such machines, the Z and W axes can be declared as carried or independent.

When Z and W are declared carried, the tool length correction is applied to only one of these two axes.

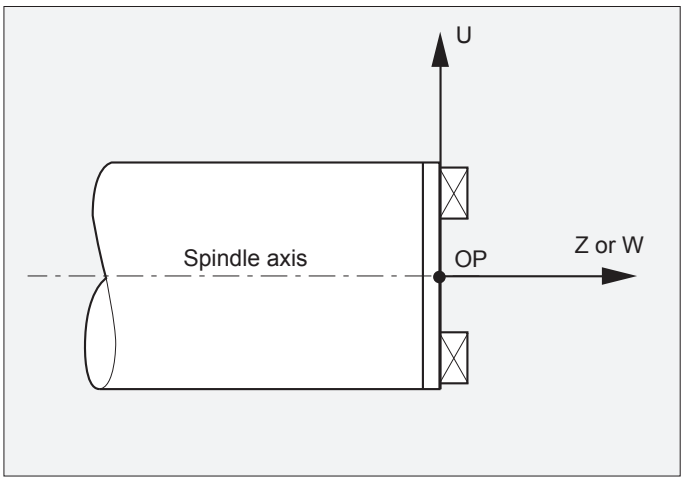
X and U Axes with Reference to Diameter or Radius

The X and U axes can be declared programmed with reference to the diameter or radius.

Programming the X and U axes with reference to diameter is used only when programming function G20 defining the ZX interpolation plane for turning.

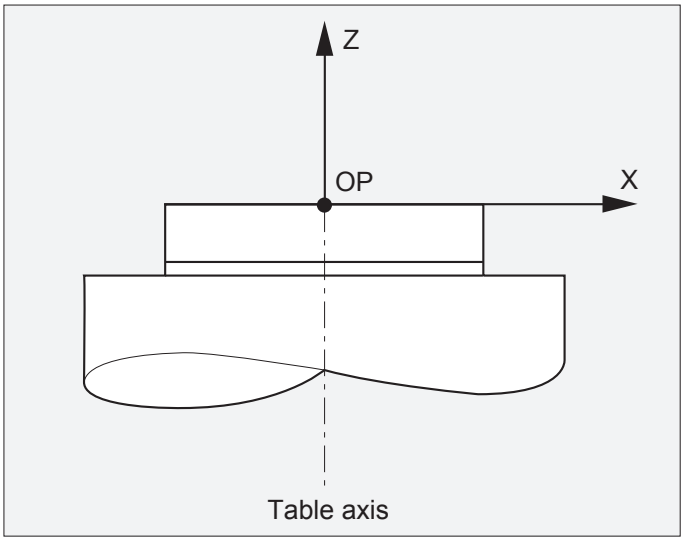
5.26.1.1 Boring Machine

The program origin (OP) for the U axis is defined in the spindle axis. Homing on the U axis is carried out in the same way as on the other axes (with DAT1 and DAT2).



5.26.1.2 Mixed Type Machine

The program origin (OP) for the X axis is defined in the table axis.



5.26.2 Assignment of Tool Dimensions to the Axes

Types of Tool Dimensions Used on Mixed Machines

- Type 0: Milling tool (dimensions D.. L.. R.. @..)
- Type 1: Turning tool (dimensions D.. X.. Z.. C..)
- Type 2: Boring tool (dimensions D.. U.. Z.. C..)

Tool Type Selection

The tool type (0, 1 or 2) is determined when entering (or loading) the tool dimensions from the keyboard, a peripheral or by programming (for additional information, refer to the Flexium HMI Operator manual M00017).

The selection of the correction type requires declaring two mandatory addresses for each tool dimension type:

- L.. and R.. for type 0 (milling)
- X.. and Z.. for type 1 (turning)
- U.. and Z.. for type 2 (boring).

Notes on Tool Dimension Assignment

Milling

The tool length L is assigned to the tool axis specified by tool orientation function G16. The tool axis must be a primary axis or a carried secondary axis. In no case can it be an independent secondary axis.

The tool radius R for radius offset (G41 or G42) is assigned to the two axes of the interpolation plane (G17, G18 or G19), be they primary, secondary, carried or independent.

Turning

The X and Z dimensions are assigned to axes X and Z or U and W if they are declared as carried (see machine parameter P64).

The tool radius R for radius offset (G41 or G42) is assigned to the two axes of the interpolation plane (G20), be they primary, secondary, carried or independent.

Boring

The U dimension is assigned only to the U axis.

The Z dimension is assigned to the Z axis or to the W axis if it is declared as carried.

The tool radius R for radius offset (G41 or G42) is assigned to the two axes of the interpolation plane (G20), be they primary, secondary, carried or independent.

Requirements for Taking Into Account the Tool Dimensions

Type 0 (milling)

The tool L, R and @ dimensions are taken into account when the type 0 correction number Dxxx and the axes concerned are programmed. The interpolation plane is G17, G18 or G19.

Type 1 (turning)

The tool X, Z and R dimensions are taken into account when the type 1 correction number Dxxx and the axes concerned are programmed. The tool nose orientation C is taken into account on the X and Z axes. The interpolation plane is G20.

Type 2 (boring)

The tool U, Z and R dimensions are taken into account when the type 2 correction number Dxxx and the axes concerned are programmed. The tool nose orientation C is taken into account on the U and Z axes. The interpolation plane is G20.

For type 1 tools (turning) or type 2 tools (boring), the tool length correction is not taken into account on the Y and V axes.

5.26.3 Preselection of the Program Origin (with function G92)

Function G92 includes an extension dedicated to boring and mixed type machines.

The axes and values programmed after the function define the current position of the tool with respect to the new program origin. DAT1 is computed according to the current tool length L; taking into account of L is therefore defined by the tool type: 0, 1 or 2.

Type 0 (milling)

L is taken into account according to the tool axis orientation function (G16 P, G16 Q or G16 R) on:

- the main axis if it is independent
- one of the axes if they are carrier/carried axis pairs.

Type 1 (turning)

L is taken into account on the X axis (or U if X and U are carried) and the Z axis (or W if Z and W are carried).

Type 2 (boring)

L is taken into account on axes U and Z (or W if Z and W are carried).

Error Messages

If the tool type is incompatible with the interpolation plane, the system returns error message 77:

- Tool type 0 (milling) with turning plane G20, G21 or G22,
- Tool type 1 (turning) or type 2 (boring) with milling plane G17, G18 or G19.

5.26.4 Notes on Programming the ISO Functions

Certain ISO functions used for milling and turning have special features, generally related to the machine axes declared. The other functions are used in the standard way.

Milling Function

G17, G18, G19: Interpolation plane selection

Possibility of various machining and contouring operations in the three main planes involving all the primary axes (X, Y, Z), secondary axes (U, V, W) and rotary axes (A, B, C).

Milling cycles

All the standard milling cycles can be used in the three interpolation planes.

Turning Functions

G20, G21, G22: Interpolation plane selection

Possibility of transforming the ZX coordinate system (G20) by programming functions G21 and G22 with the rotary C axis (do not confuse the spindle with the rotary C axis). Programming of G20, G21 or G22 sets function G18 (ZX plane for milling).

Turning cycles

The following cycles can only be used in the ZX plane (G20):

- G63: Roughing cycle with groove.
- G64: Turn/face roughing cycle.
- G65: Groove roughing cycle.
- G66: Plunging cycle.
- G33: Constant lead thread cutting. Possible on X or U or Z or W axes.
- G38: Sequenced thread cutting. Possible on X or U or Z or W axes.

Common Turning and Milling Functions

M62, M63, M64, M65: Spindle control

Possibility of controlling the spindles simultaneously or not by choosing the spindle number to be controlled (1 to 4).

M66, M67, M68, M69: Spindle measurement

Possibility of measuring the spindles by selecting the spindle number to be measured (1 to 4).

G94 or G95: Feed rate

Possible feed rate in mm/min (G94) or mm/rev (G95) on all the linear axes (X, Y, Z, U, V, W) and selection of the measured spindle in G95 (see M66 to M69).

G96: Constant surface speed in m/min

Possibility of defining the speed of one of the spindles (see M66 to M69) according to the position of the X or U axis. This function is only active in planes G18 (milling) or G20 (turning).

The choice of the X or U axis is related to the type of correction selected (1 or 2).

Page deliberately empty

Summary

6 Profile Geometry Programming 421
6.1 Profile Geometry.....421
6.1.1 Definition of the Addresses Characterising PGP.....422
6.1.1.1 Addresses with Values ► X../Y.. , Z.., EA, X.. Y.., I.. , K.., J.., R.., EB+, EB- 422
6.1.1.2 Addresses without Values ► ET, ES, E+/E- 425
6.1.2 Programming of Geometric Elements 427
6.1.2.1 Programming of Fully Defined Geometric Elements 427
6.1.2.2 Programming of Geometric Elements not Completely Defined.....429
6.1.3 Programming Chamfers and Fillets Between Two Elements 443

Page deliberately empty

6 Profile Geometry Programming

6.1 Profile Geometry

General

The system gives the user the possibility of programming all or part of a profile consisting of geometric elements.

The system computes the undefined connecting and intersection points between elements located in the same plane.

The points can be computed between the following elements:
- line/line,
- line/circle,
- circle/circle.

Profile Geometry Programming (PGP):
- can coexist with ISO programming,
- can only be used with absolute dimensions (G90),
- applies to the plane selected XY, ZX, YZ for Milling Machines by function G17, G18 or G19 (the change of plane must be programmed on a fully defined point), for Turning Machines it applies for interpolation planes ZX, XY, YZ (by function G20, G21 or G22)
- allows programming of the tool axis (Z in the XY plane) in a block.

Definition of the Geometric Elements

Profile Geometry Programming (PGP) is carried out by writing a sequence of blocks.

Each block includes a geometric element which can be:
- a line segment, or
- an arc.

A geometric element can be completely or incompletely defined in a block.

Completely defined elements are :
- the end point of a line,
- the end point of an arc and the coordinates of the centre or the radius.

If an element is incompletely defined, then additional information must be contained in the next block or the next two blocks (excluding fillets and/or chamfers).

The necessary and sufficient blocks allowing the system to compute all the coordinates of a geometric element is called a «geometric entity» (see definition).

The origin of a geometric entity is the start point of the first element.

The start point is:
- programmed in the previous block, or
- already computed by the system (the first block of an entity can also be the last block of the previous entity).

Definition of an Entity

A PGP geometric entity is a self-sufficient part of a profile.

Most following examples are given in the XY plane for Milling Machine system. They apply for all couple of interpolable axes for Turning or Milling Machine systems as well.

6.1.1 Definition of the Addresses Characterising PGP

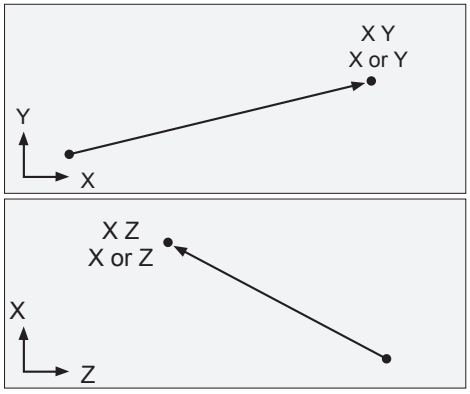
The following PGP addresses are processed in the XY plane (G17).
For the other planes (ZX and YZ), use the axes of the plane selected.

6.1.1.1 Addresses with Values ► X../Y.. , Z.., EA, X.. Y.., I.. , K.., J.., R.., EB+, EB-

X../Y.., Z.. or X.. Y.., Z..

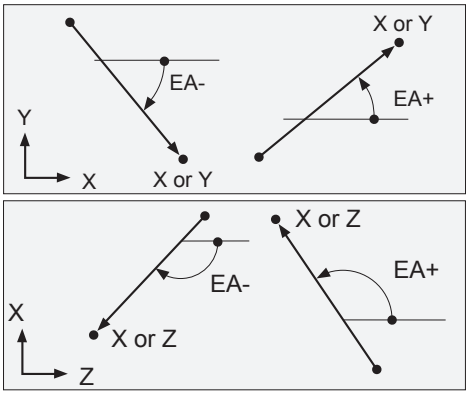
X../Y..,Z.., or X.. Y.., Z.., :

Coordinates of the end point of a line



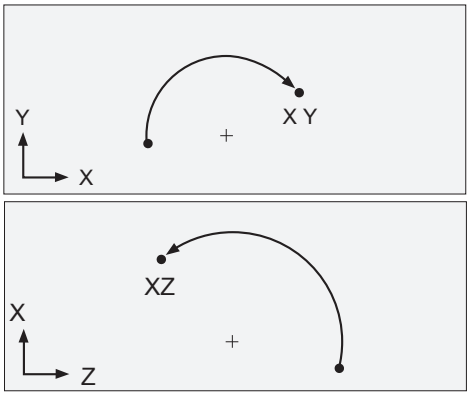
EA..

EA.. : **Angle element of a line**



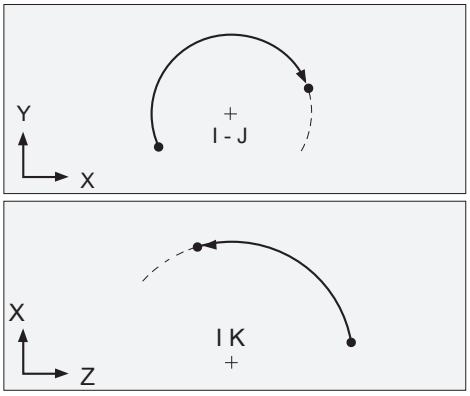
X.. Y.. Z..

X.. Y.. Z..: **Coordinates of the end point of a circle**



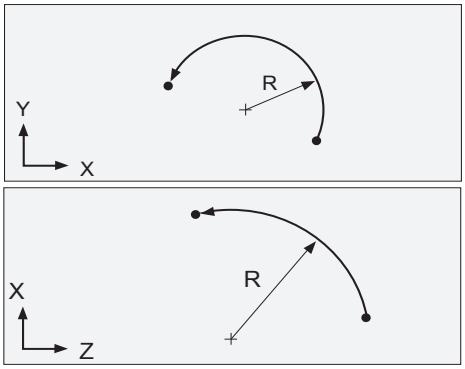
I.. J.. K..

I.. J.. K.. : **Coordinates of the centre of a circle**



R..

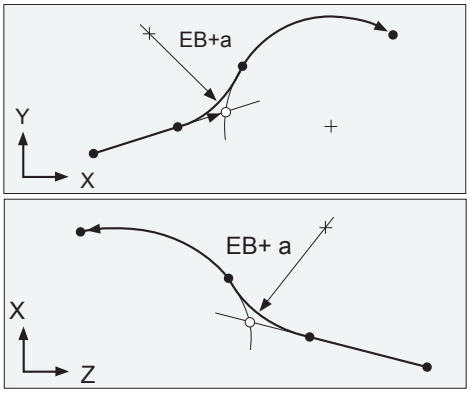
R.. : **Radius of a circle**



EB+..

EB+.. : **Fillet between two intersecting elements (e.g. line/circle)**

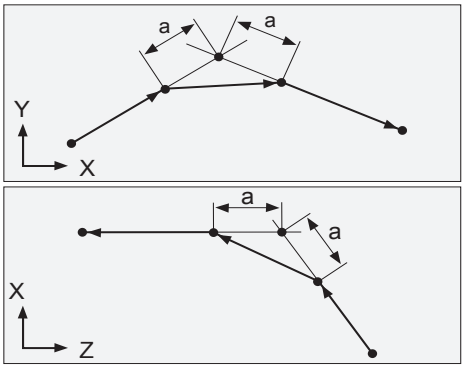
The block containing EB+.. and the next block are connected by a fillet (a=value programmed with EB+).



EB-..

EB-.. : **Chamfer between two intersecting lines (only)**

The block containing EB-.. and the next block are connected by a chamfer (a=value programmed with EB-).

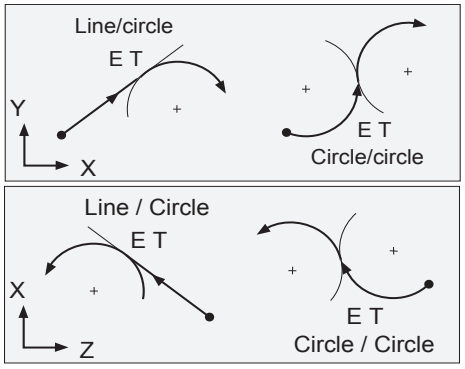


6.1.1.2 Addresses without Values ► ET, ES, E+/E-

ET

ET : Tangent element

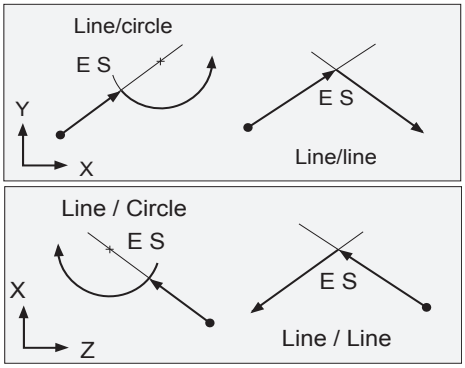
The block containing ET and the next block are tangent. ET is optional unless it is the only function characterising the element, in which case it is mandatory (see chapter 6.1.2.2, Figures 10 and 14).



ES

ES : Secant element

The block containing ES and the next block intersect. If the intersection point of two secant elements is not programmed, ES must be specified in the first block.



E+/E-

E+/E- : Discriminant

The discriminant is used to dispel the ambiguity when the programming of one or more blocks leaves a choice between two possible solutions.

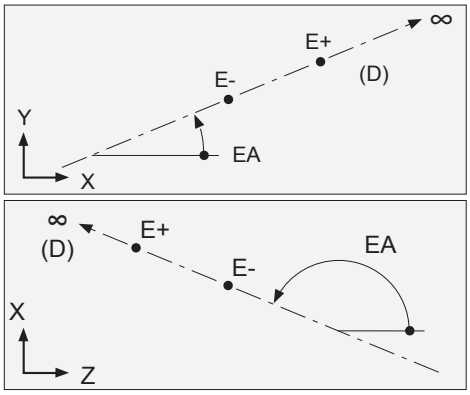
When the discriminant determines an element of an entity:
- it must be programmed in the first block of this entity,
- the + sign or - sign specifies the position of a characteristic point of one of the two solutions with respect to an imaginary line pointing in direction D.

The characteristic points can be:
- the intersection point between two secant elements,
- the point of tangency between two elements,
- the position of the centre of a circle.

The pointing line (D) is:
- the line defined by angle EA.. (if one of the elements of the entity is thus defined),
- the line connecting a known point of the first element to a known point of the last element of the entity (pointing from the first to the last). This known point is preferably the centre of a circle programmed by I and J or, lacking such, another programmed point.

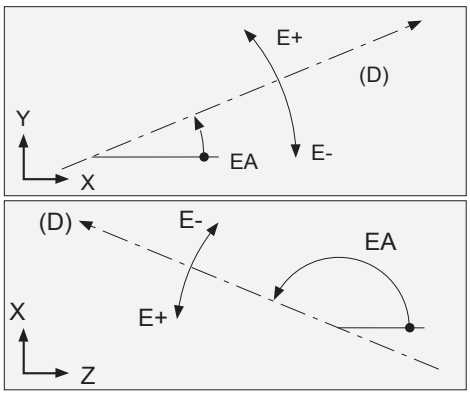
Characteristic points located on the pointing line (D)

- E+ defines the point closest to + ∞ (plus infinity) on line D.
- E- defines the point closest to - ∞ (minus infinity) on line D.



Characteristic points on either side of the pointing line (D)

- E+ defines a point to left of line D.
- E- defines a point to the right of D.



The discriminant can be combined with address ES (secant element) and address ET (tangent element). Example:

- E± secant is programmed as ES+ or ES- .
- E± tangent is programmed as ET+ or ET- .

Programming of the discriminant with secant elements

For line/circle and circle/circle intersecting elements, the system will compute two possible solutions. Programming of the discriminant with ES (ES+ or ES-) is therefore mandatory (see Sec. 6.1.2.2, figures 3a and 3b for instance).

Programming the discriminant with tangent elements

The system allows only two possible solutions (only continuous tangencies without backtracking are allowed by the system).

When there are two possible solutions, these solutions result in:
- the creation of an arc less than 180 degrees, and
- the creation of an arc greater than 180 degrees.

In the two cases, programming of the discriminant with ET is optional. The default solution chosen by the system is that with the smallest arc (see chapter 6.1.2.2, Figures 8a and 8b, for instance).

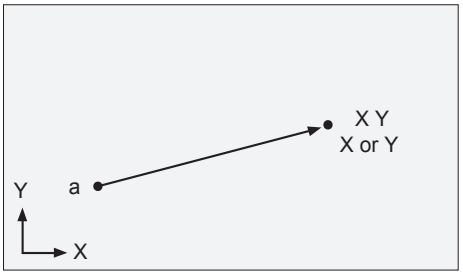
Only exception:

Circle whose centre is inside the next circle and characterised only by: the coordinates of this centre and the fact as it is tangent to the next circle (see Sec. 6.1.2.2, Figures 5b, 12b, 23b).

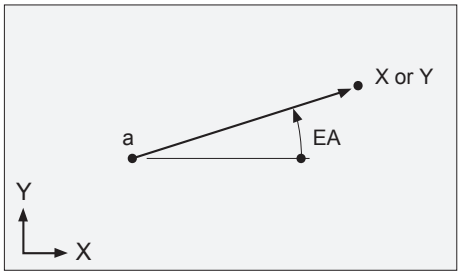
6.1.2 Programming of Geometric Elements

6.1.2.1 Programming of Fully Defined Geometric Elements

Fully defined linear geometric element (point «a» defined)

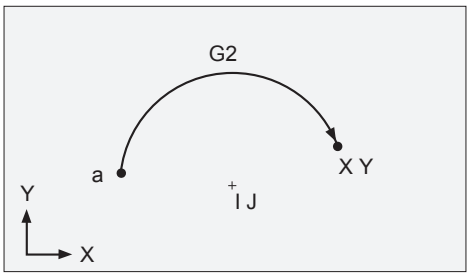



```
   N.. G01 X..
or N.. G01 Y..
or N.. G01 X.. Y..
```

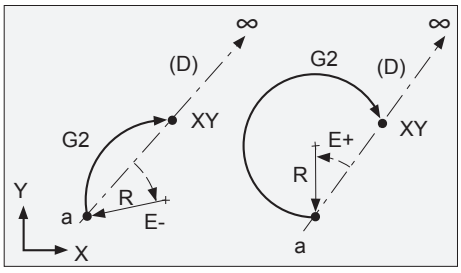



```
   N.. G01 EA.. X..
or N.. G01 EA.. Y..
```

Fully defined circular geometric element (point «a» defined)

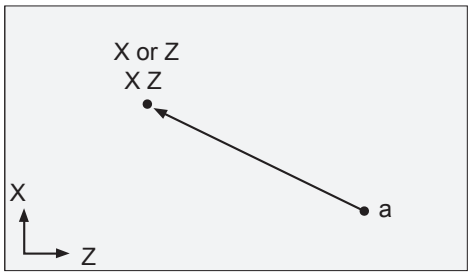



```
N.. G02/G03 X.. Y.. I.. J..
```

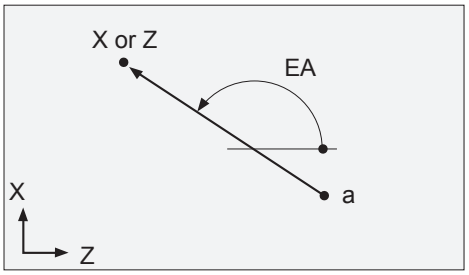



```
N.. G02/G03 X.. Y.. R.. E+/E-
```

Fully defined linear geometric element (point «a» defined)

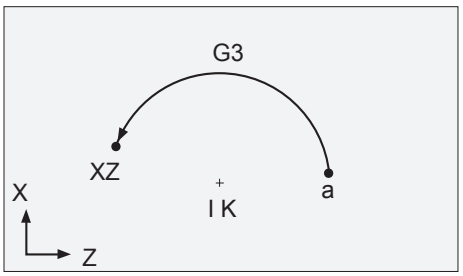



```
   N.. G01 X..
or N.. G01 Z..
or N.. G01 X.. Z..
```

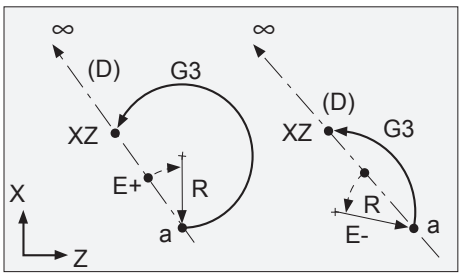



```
   N.. G01 EA.. X..
or N.. G01 EA.. Z..
```

Fully defined circular geometric element (point «a» defined)




```
N.. G02/G03 X.. Z.. I.. K..
```



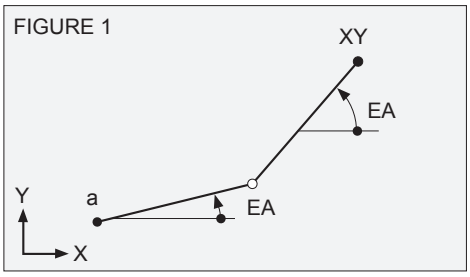

```
N.. G02/G03 X.. Z.. R.. E+/E-
```

6.1.2.2 Programming of Geometric Elements not Completely Defined

Geometric elements defined using the following block(s)

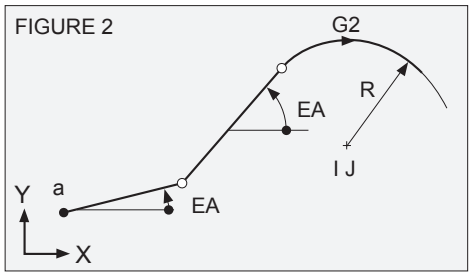
The first element is a line (start point «a» is fully defined).

FIGURE 1



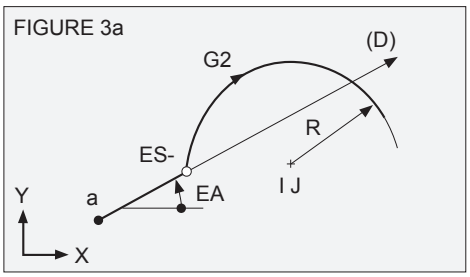

```
N.. G01 EA.. ES
N.. EA X.. Y..
```

FIGURE 2



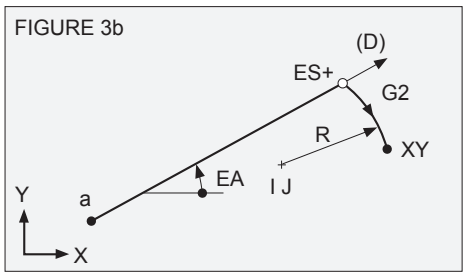

```
N.. G01 EA.. ES
N.. EA..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 3a



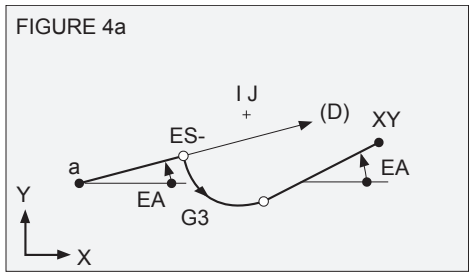

```
N.. G01 EA.. ES-
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 3b



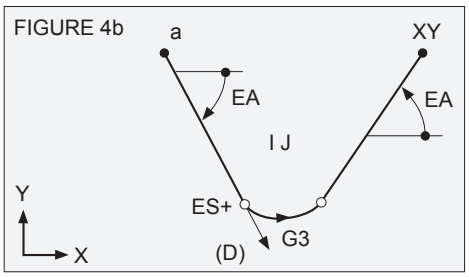

```
N.. G01 EA.. ES+
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 4a

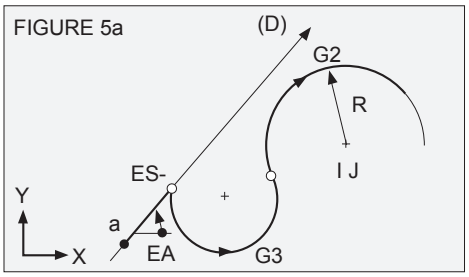



```
N.. G01 EA.. ES-
N.. G02/G03 I..J..
N.. G01 EA.. X.. Y..
```

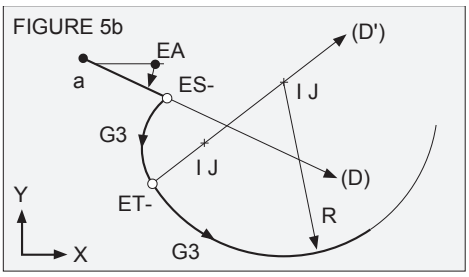
FIGURE 4b



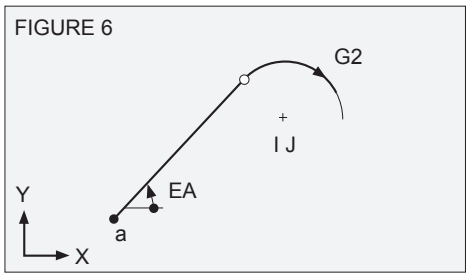

```
N.. G01 EA.. ES+
N.. G02/G03 I..J..
N.. G01 EA.. X.. Y..
```



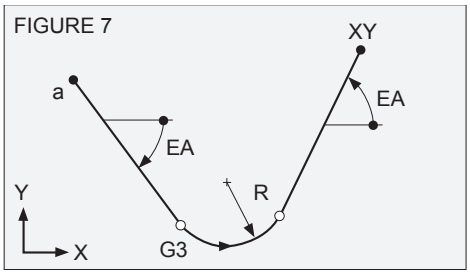

```
N.. G01 EA.. ES-
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..
```



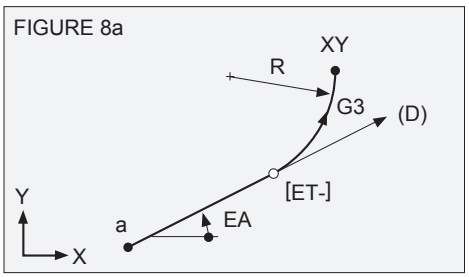

```
N.. G01 EA.. ES-
N.. G02/G03 I.. J.. ET-
N.. G02/G03 I.. J.. R../X.. Y..
```



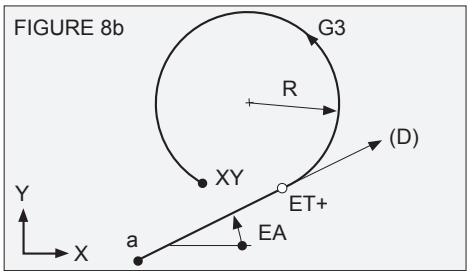

```
N.. G01 EA..
N.. G02/G03 I.. J..
```




```
N.. G01 EA..
N.. G02/G03 R..
N.. G01 EA.. X.. Y..
```

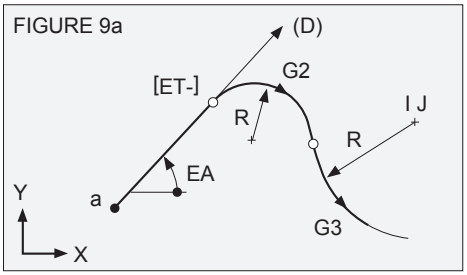



```
N.. G01 EA..
N.. G02/G03 R.. X.. Y..
```



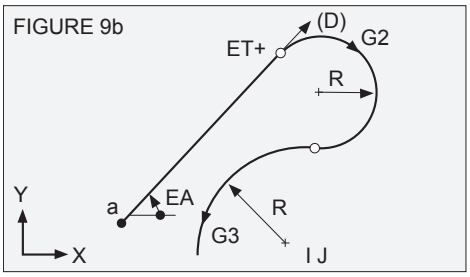

```
N.. G01 EA.. ET+
N.. G02/G03 R.. X.. Y..
```

FIGURE 9a



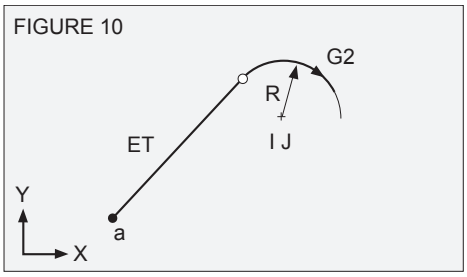

```
N.. G01 EA..
N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 9b



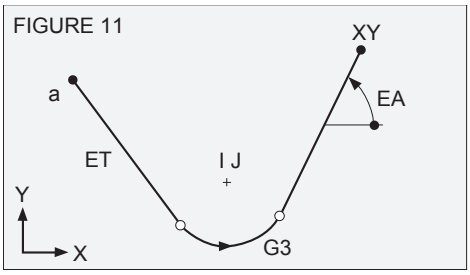

```
N.. G01 EA.. ET+
N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 10



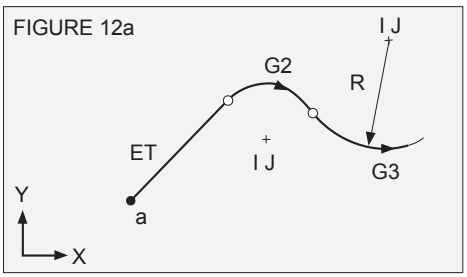

```
N.. G01 ET
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 11



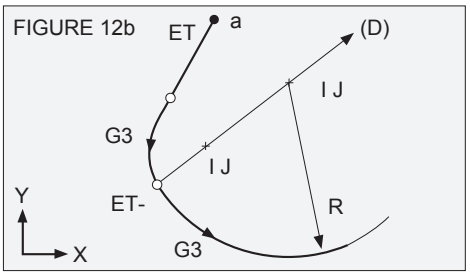

```
N.. G01 ET
N.. G02/G03 I.. J..
N.. G01 EA.. X.. Y..
```

FIGURE 12a




```
N.. G01 ET
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..
```

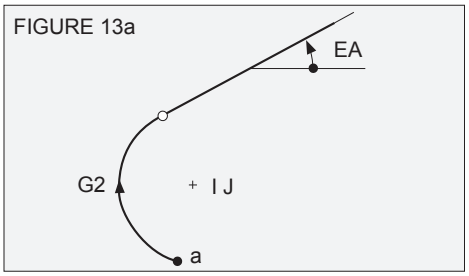
FIGURE 12b




```
N.. G01 ET
N.. G02/G03 I.. J.. ET-
N.. G02/G03 I.. J.. R../X.. Y..
```

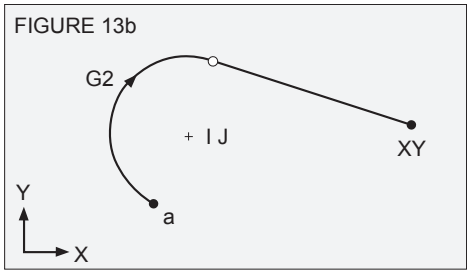
The first element is a circle (start point «a» is fully defined).

FIGURE 13a



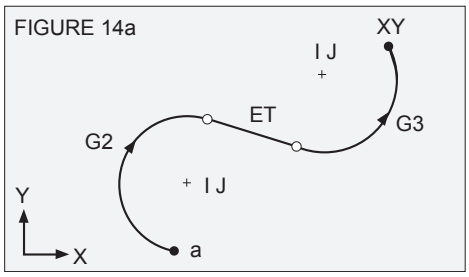

```
N.. G02/G03 I.. J..
N.. G01 EA..
```

FIGURE 13b



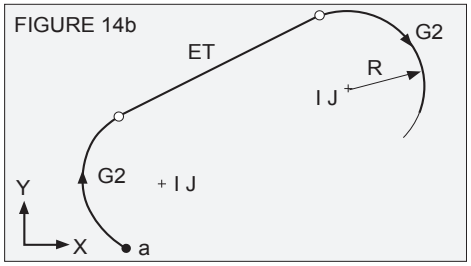

```
N.. G02/G03 I.. J..
N.. G01 X.. Y..
```

FIGURE 14a



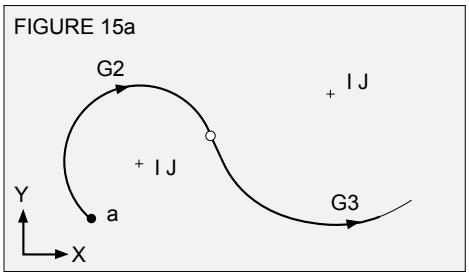

```
N.. G02/G03 I.. J..
N.. G01 ET
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 14b



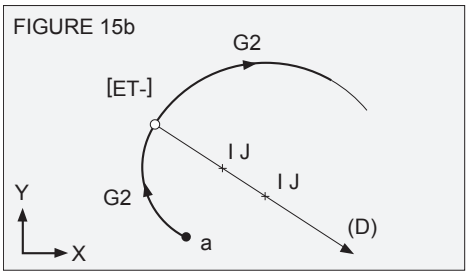

```
N.. G02/G03 I.. J..
N.. G01 ET
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 15a



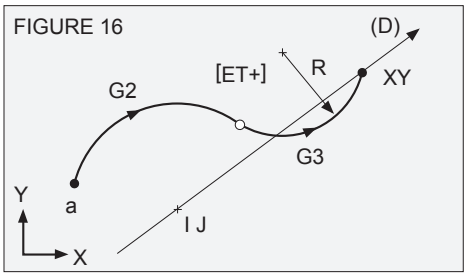

```
N.. G02/G03 I.. J..
N.. G02/G03 I.. J..
```

FIGURE 15b



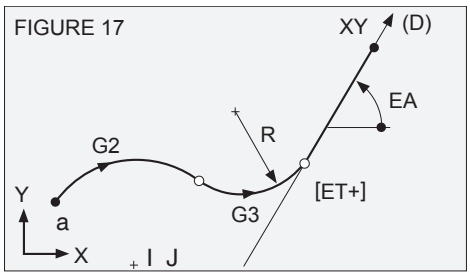

```
N.. G02/G03 I.. J..
N.. G02/G03 I.. J..
```

FIGURE 16



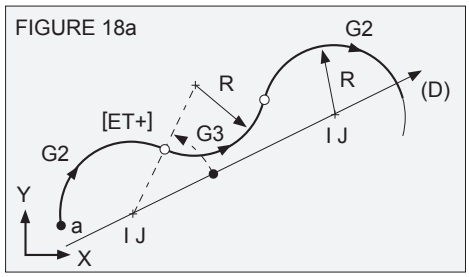

```
N.. G02/G03 I.. J..
N.. G02/G03 R.. X.. Y..
```

FIGURE 17



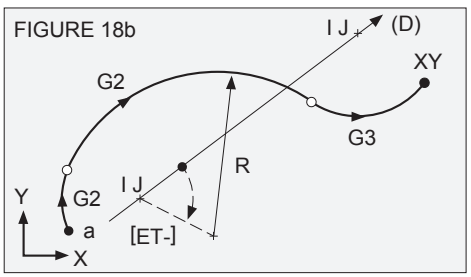

```
N.. G02/G03 I.. J..
N.. G02/G03 R..
N.. G01 EA.. X.. Y..
```

FIGURE 18a



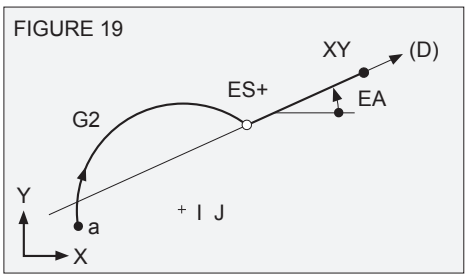

```
N.. G02/G03 I.. J..
N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 18b



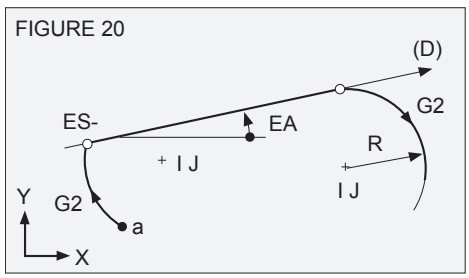

```
N.. G02/G03 I.. J..
N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 19



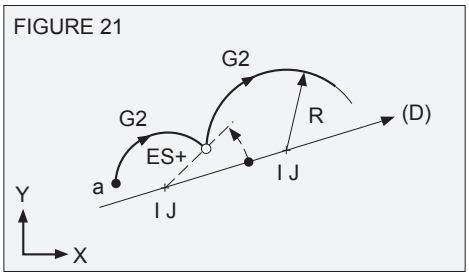

```
N.. G02/G03 I.. J.. ES+
N.. G01 EA.. X.. Y..
```

FIGURE 20



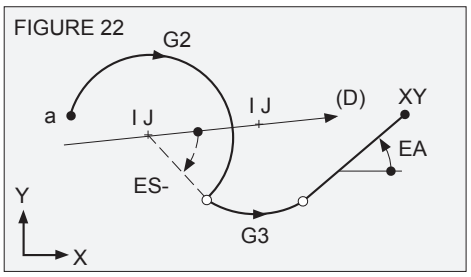

```
N.. G02/G03 I.. J.. ES-
N.. G01 EA..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 21



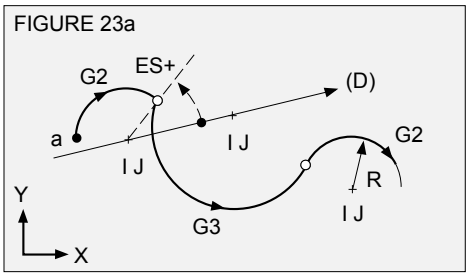

```
N.. G02/G03 I.. J.. ES+
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 22



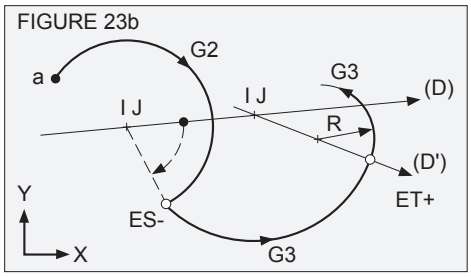

```
N.. G02/G03 I.. J.. ES-
N.. G02/G03 I.. J..
N.. G01 EA X.. Y..
```

FIGURE 23a



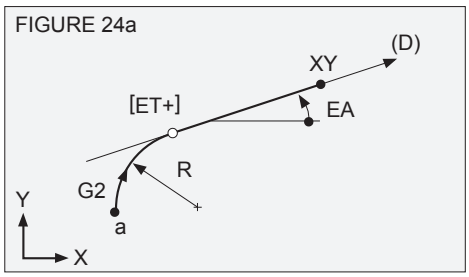

```
N.. G02/G03 I.. J.. ES+
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 23b



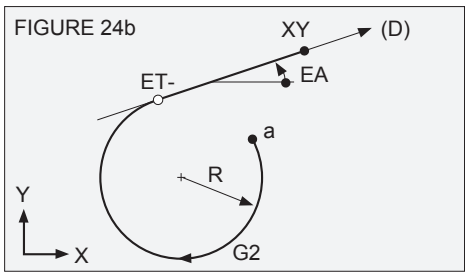

```
N.. G02/G03 I.. J.. ES-
N.. G02/G03 I.. J.. ET+
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 24a



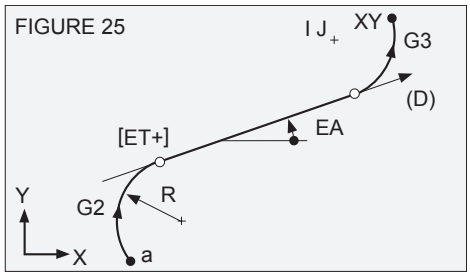

```
N.. G02/G03 R..
N.. G01 EA.. X.. Y..
```

FIGURE 24b



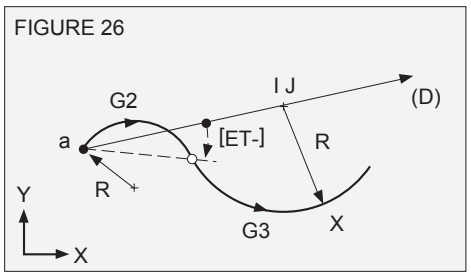

```
N.. G02/G03 R.. ET-
N.. G01 EA.. X.. Y..
```

FIGURE 25



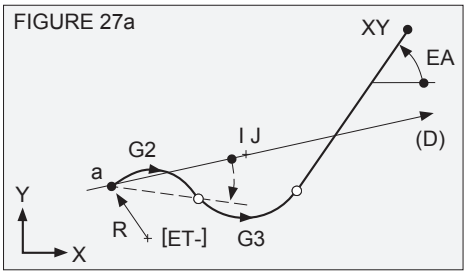

```
N.. G02/G03 R..
N.. G01 EA..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 26



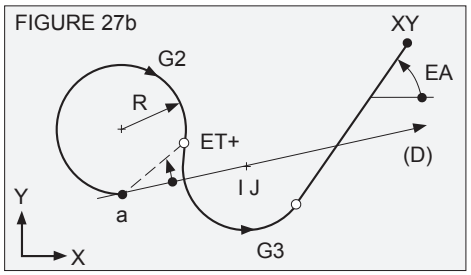

```
N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 27a



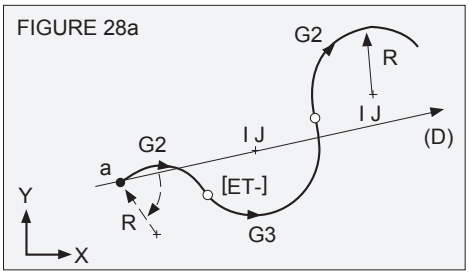

```
N.. G02/G03 R..
N.. G02/G03 I.. J..
N.. G01 EA.. X.. Y..
```

FIGURE 27b



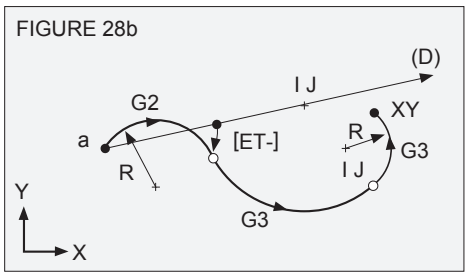

```
N.. G02/G03 R.. ET+
N.. G02/G03 I.. J..
N.. G01 EA.. X.. Y..
```

FIGURE 28a




```
N.. G02/G03 R..
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..
```

FIGURE 28b

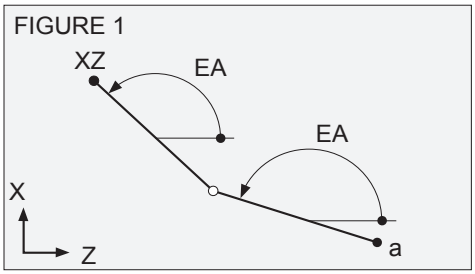



```
N.. G02/G03 R..
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..
```

Geometric elements defined using the following block(s)

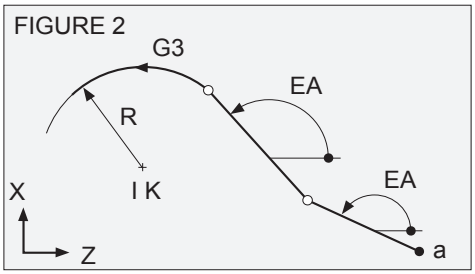
The first element is a line (start point «a» is fully defined).

FIGURE 1



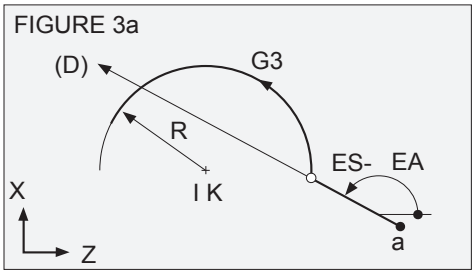

```
N.. G01 EA.. ES
N.. EA X.. Z..
```

FIGURE 2



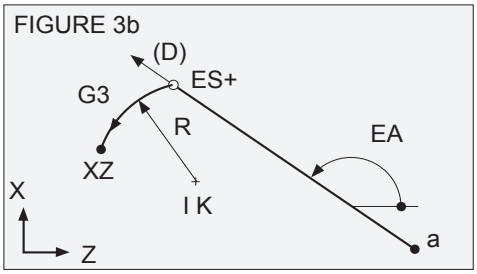

```
N.. G01 EA.. ES
N.. EA..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 3a



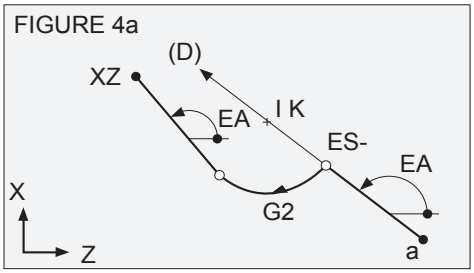

```
N.. G01 EA.. ES-
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 3b



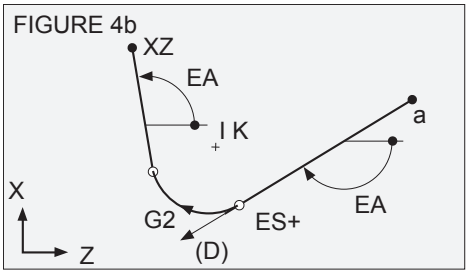

```
N.. G01 EA.. ES+
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 4a



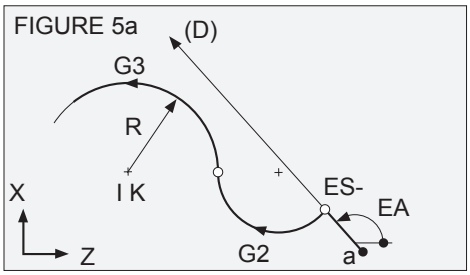

```
N.. G01 EA.. ES-
N.. G02/G03 I..K..
N.. G01 EA.. X.. Z..
```

FIGURE 4b



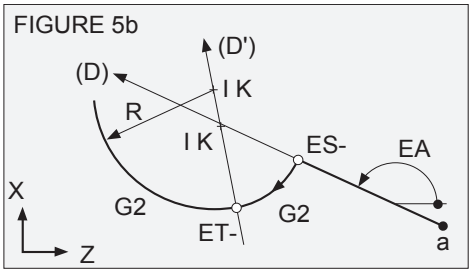

```
N.. G01 EA.. ES+
N.. G02/ G03 I..K..
N.. G01 EA.. X.. Z..
```

FIGURE 5a



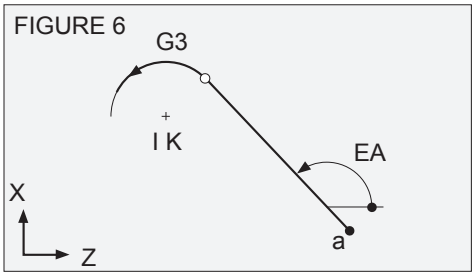

```
N.. G01 EA.. ES-
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 5b



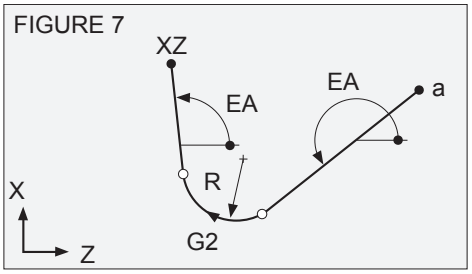

```
N.. G01 EA.. ES-
N.. G02/G03 I.. K.. ET-
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 6



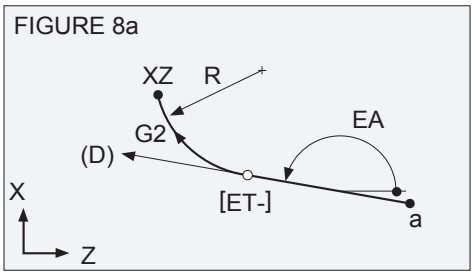

```
N.. G01 EA..
N.. G02/G03 I.. K..
```

FIGURE 7



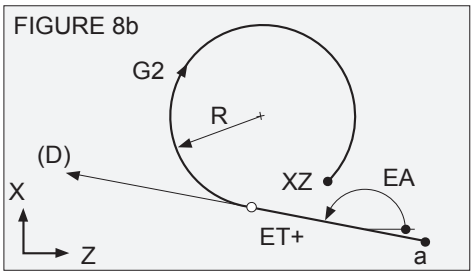

```
N.. G01 EA..
N.. G02/G03 R..
N.. G01 EA.. X.. Z..
```

FIGURE 8a

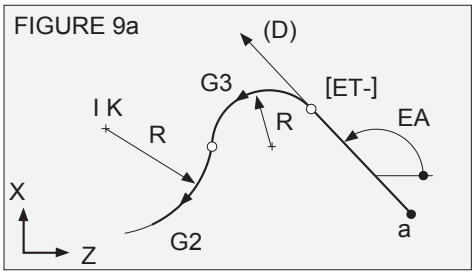



```
N.. G01 EA..
N.. G02/G03 R.. X.. Z..
```

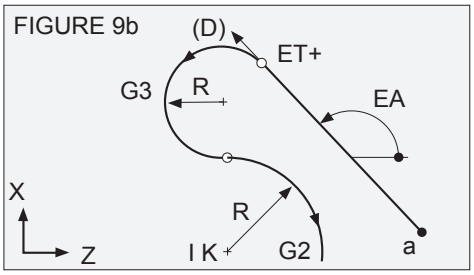
FIGURE 8b



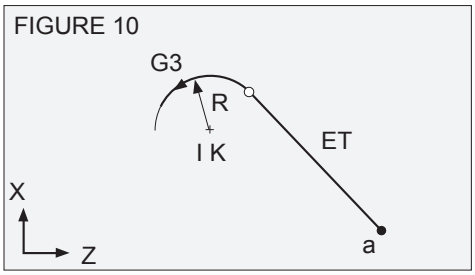

```
N.. G01 EA.. ET+
N.. G02/G03 R.. X.. Z..
```



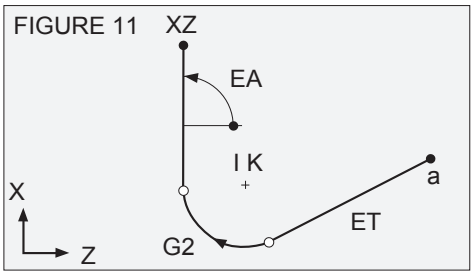

```
N.. G01 EA..
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..
```



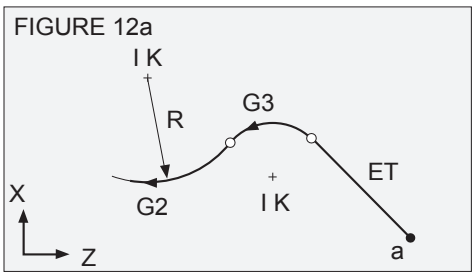

```
N.. G01 EA.. ET+
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..
```



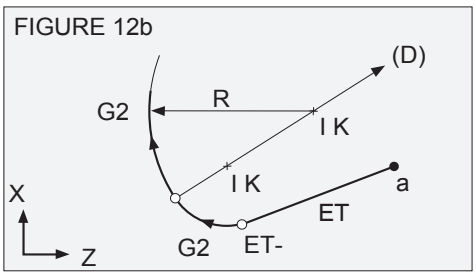

```
N.. G01 ET
N.. G02/G03 I.. K.. R../X.. Z..
```




```
N.. G01 ET
N.. G02/G03 I.. K..
N.. G01 EA.. X.. Z..
```



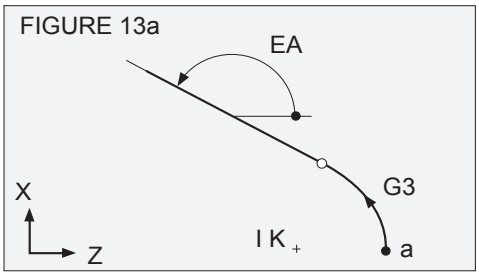

```
N.. G01 ET
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```




```
N.. G01 ET
N.. G02/G03 I.. K.. ET-
N.. G02/G03 I.. K.. R../X.. Z..
```

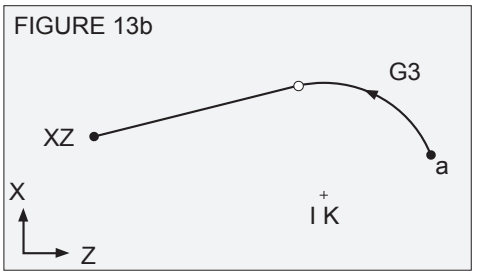
The first element is a circle (start point «a» is fully defined).

FIGURE 13a



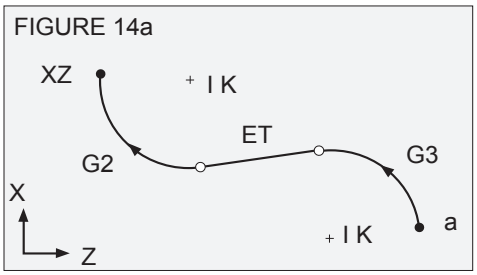

```
N.. G02/G03 I.. K..
N.. G01 EA..
```

FIGURE 13b



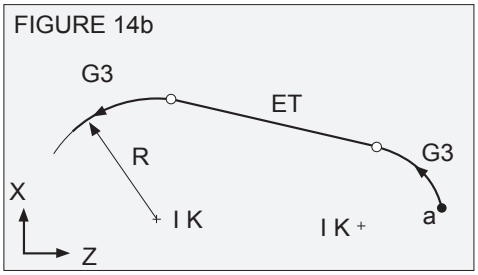

```
N.. G02/G03 I.. K..
N.. G01 X.. Z..
```

FIGURE 14a



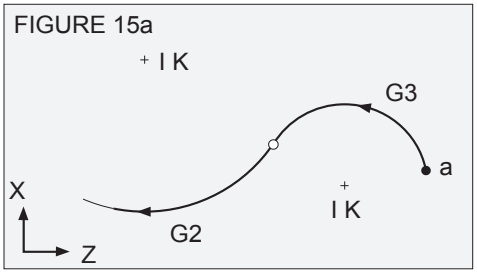

```
N.. G02/G03 I.. K..
N.. G01 ET
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 14b



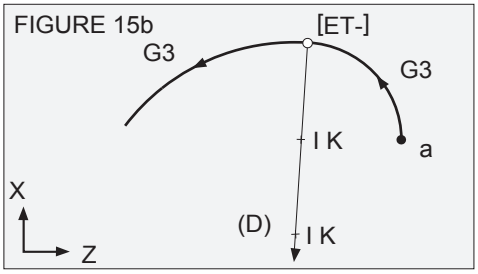

```
N.. G02/G03 I.. K..
N.. G01 ET
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 15a



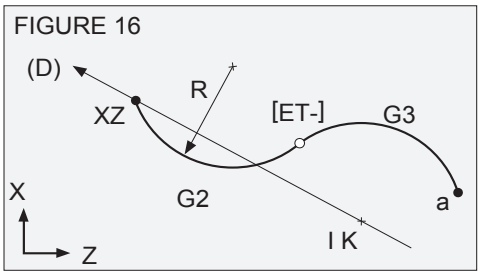

```
N.. G02/G03 I.. K..
N.. G02/G03 I.. K..
```

FIGURE 15b



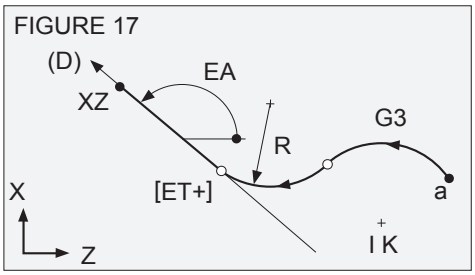

```
N.. G02/G03 I.. K..
N.. G02/G03 I.. K..
```

FIGURE 16



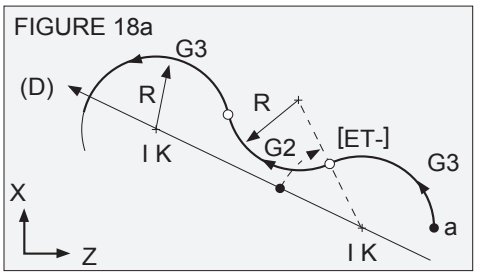

```
N.. G02/G03 I.. K..
N.. G02/G03 R.. X.. Z..
```

FIGURE 17



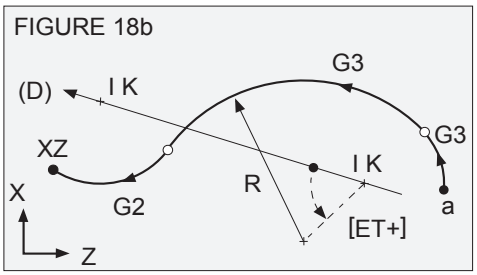

```
N.. G02/G03 I.. K..
N.. G02/G03 R..
N.. G01 EA.. X.. Z..
```

FIGURE 18a



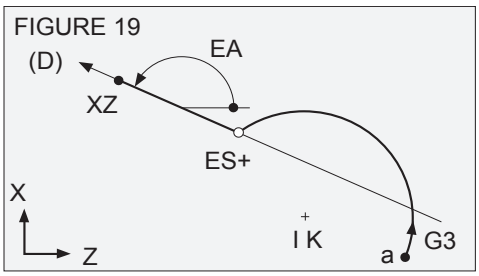

```
N.. G02/G03 I.. K..
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 18b



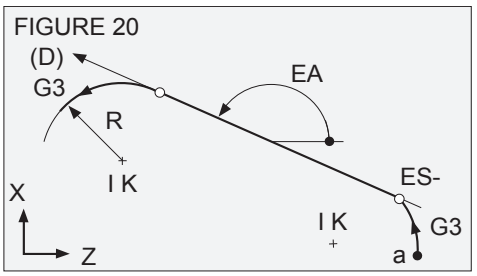

```
N.. G02/G03 I.. K..
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 19



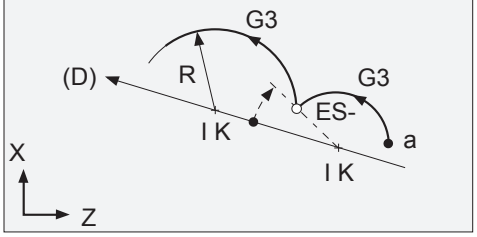

```
N.. G02/G03 I.. K.. ES+
N.. G01 EA.. X.. Z..
```

FIGURE 20



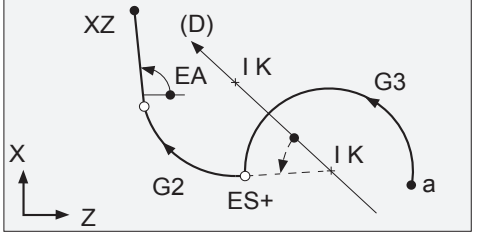

```
N.. G02/G03 I.. K.. ES-
N.. G01 EA..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 21



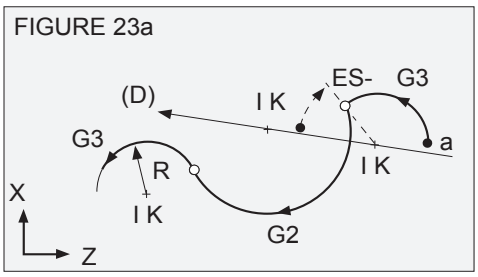

```
N.. G02/G03 I.. K.. ES-
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 22



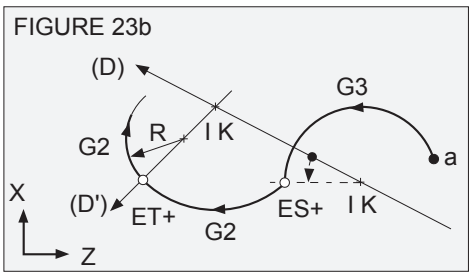

```
N.. G02/G03 I.. K.. ES+
N.. G02/G03 I.. K..
N.. G01 EA X.. Z..
```

FIGURE 23a



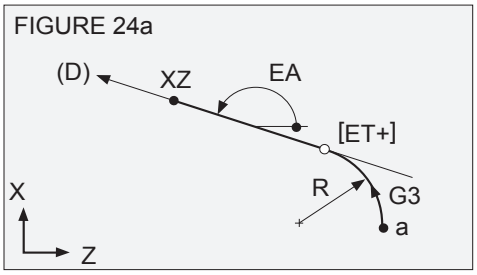

```
N.. G02/G03 I.. K.. ES-
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 23b



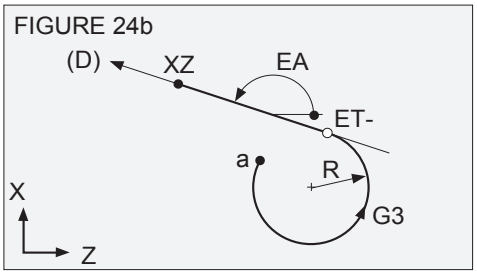

```
N.. G02/G03 I.. K.. ES+
N.. G02/G03 I.. K.. ET+
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 24a



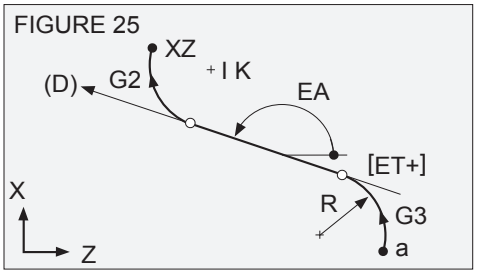

```
N.. G02/G03 R..
N.. G01 EA.. X.. Z..
```

FIGURE 24b



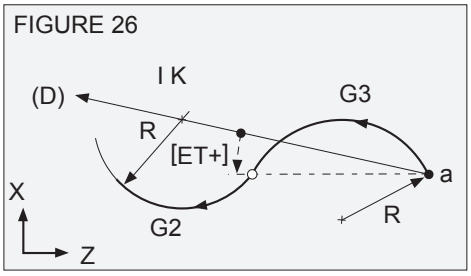

```
N.. G02/G03 R.. ET-
N.. G01 EA.. X.. Z..
```

FIGURE 25



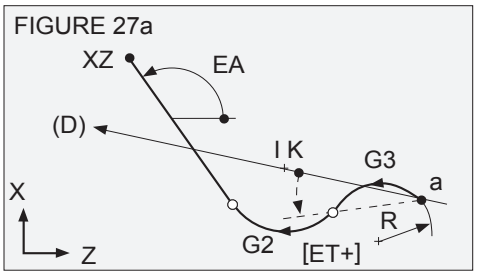

```
N.. G02/G03 R..
N.. G01 EA..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 26



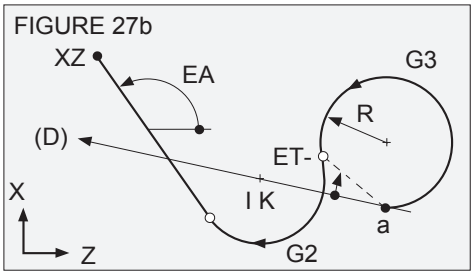

```
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 27a



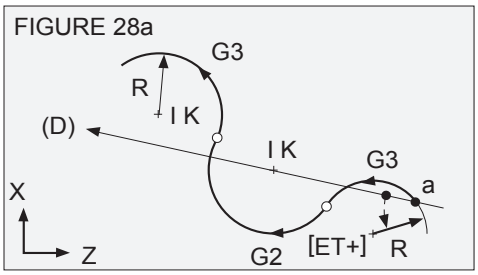

```
N.. G02/G03 R..
N.. G02/G03 I.. K..
N.. G01 EA.. X.. Z..
```

FIGURE 27b



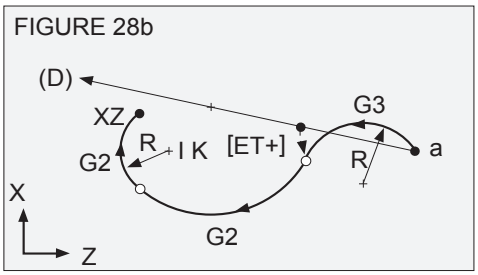

```
N.. G02/G03 R.. ET-
N.. G02/G03 I.. K..
N.. G01 EA.. X.. Z..
```

FIGURE 28a



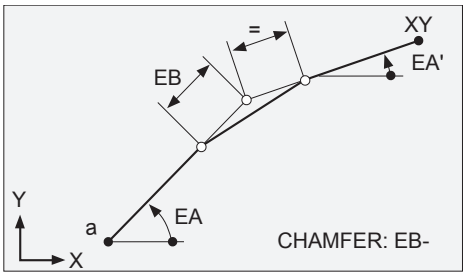

```
N.. G02/G03 R..
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 28b

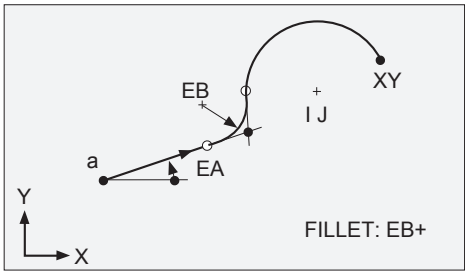



```
N.. G02/G03 R..
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```

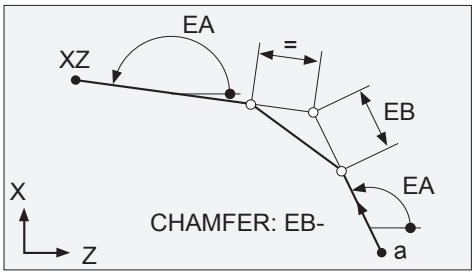
6.1.3 Programming Chamfers and Fillets Between Two Elements



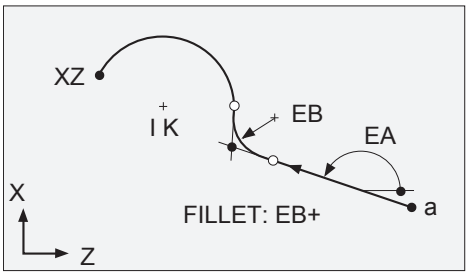

```
N.. G01 EA.. ES EB-..
N.. G01 EA.. X.. Y..
```




```
N.. G01 EA.. ES- EB+..
N.. G02/G03 I.. J.. X.. Y..
```



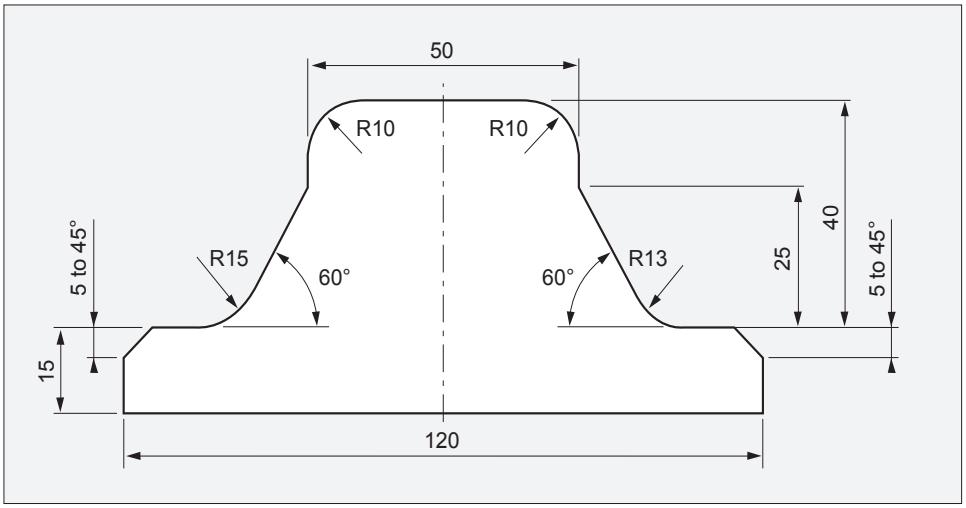

```
N.. G01 EA.. ES EB-..
N.. G01 EA.. X.. Z..
```



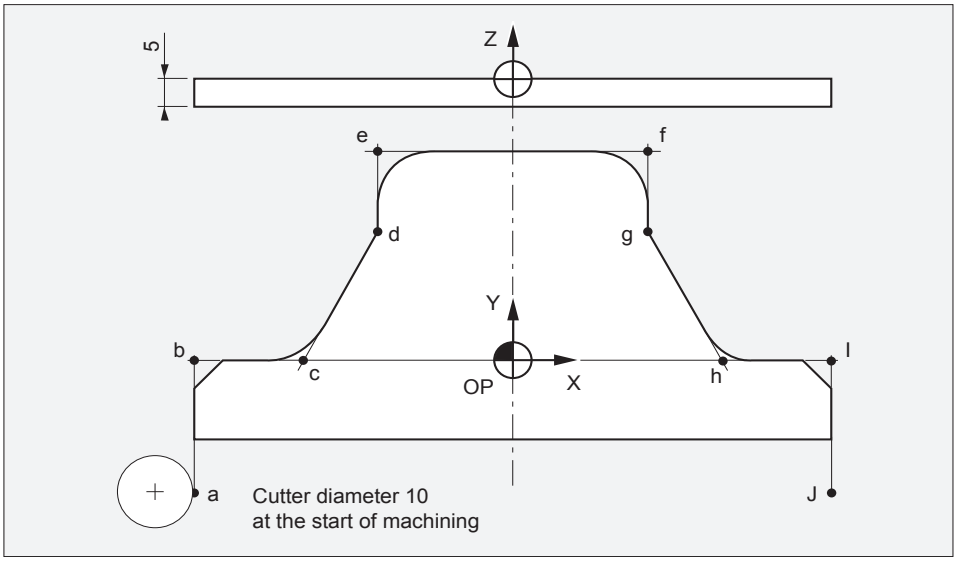

```
N.. G01 EA.. ES- EB+..
N.. G02/G03 I.. K.. X.. Z..
```

Examples of Programming in PGP

Milled part #1 profile definition



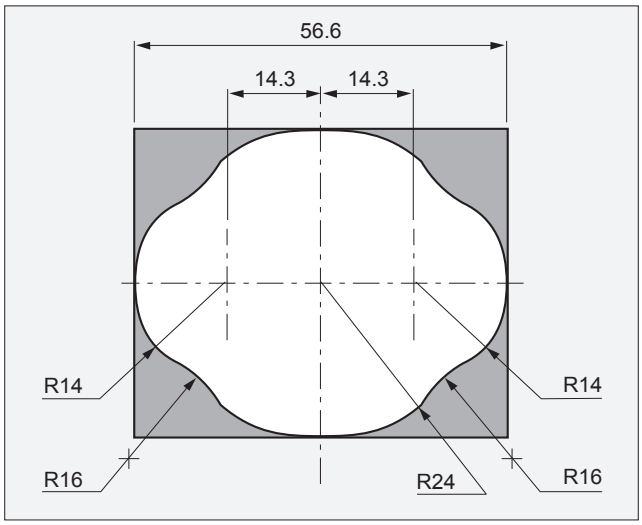
Machining paths



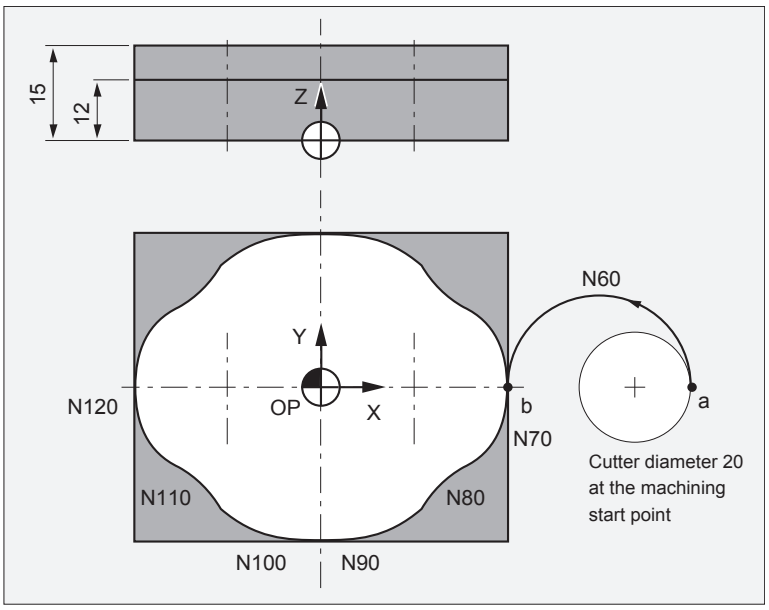

```
%61
N10 G00 G52 Z0
N20 T03 D03 M06 (CUTTER DIAMETER=10)
N30 S600 M40 M03
N35 G92 R10
N40 G00 G41 X-60 Y-25                Point a, approach
N50 Z-6                              Position on Z
N60 G01 Y0 EB-5 F150 M08             Point b
N70 EA0 ES EB15                      Point c
N80 EA60 X-25 Y25                    Point d
N90 Y40 EB10                         Point e
N100 X25 EB10                        Point f
N110 Y25                             Point g
N120 EA-60 Y0 EB13                   Point h
N130 X60 EB-5                        Point i
N140 Y-20                            Point j
N150 G00 G40 G52 Z0 M05 M09          Retraction
N160 M02
```

Half part profile in XY plane

Use of the mirror function to machine the other half-profile in the XY plane (G17).

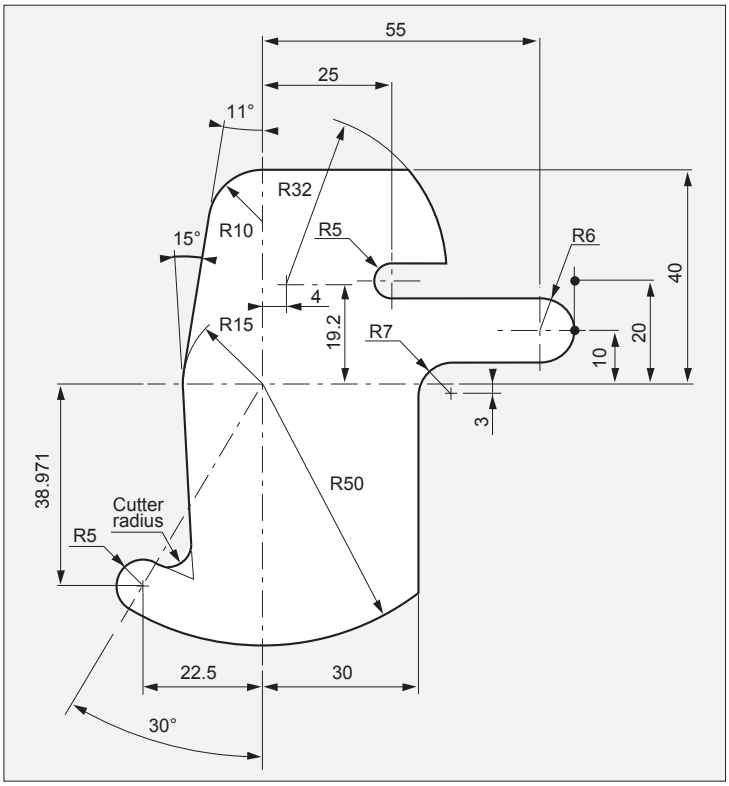


Machining paths

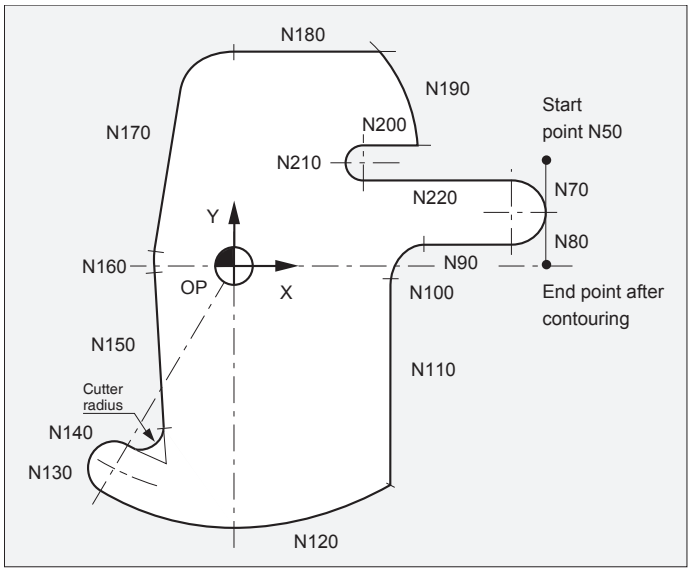



```
%3
N10 G00 G52 Z0
N20 T01 D01 M06 (CUTTER DIAMETER=20)
N30 S280 M40 M03
N40 G00 G41 X60 Y0                    Point a, approach
N50 Z12 M08                           Position on Z
N55 G92 R10                           Limit value of tangential feed rate
N60 G03 X28.3 Y0 I44.15 J0 F200       Point b
N70 G02 I14.3 J0 R14 ET               Machining of the first half-profile
N80 G03 R16 ET
N90 G02 X0 Y-24 I0 J0
N100 G02 I0 J0 R24 ET
N110 G03 R16 ET
N120 G02 X-28.3 Y0 I-14.3 J0
N130 G03 X-60 Y0 I-44.15 J0
N140 G51 X- Y-                        Mirror
N150 G77 N60 N120                     Machining of the second half-profile
N160 G51 X+ Y+
N170 G00 G40 G52 Z0 M05 M09
N180 M02
```

Milled Part#2 Profile definition



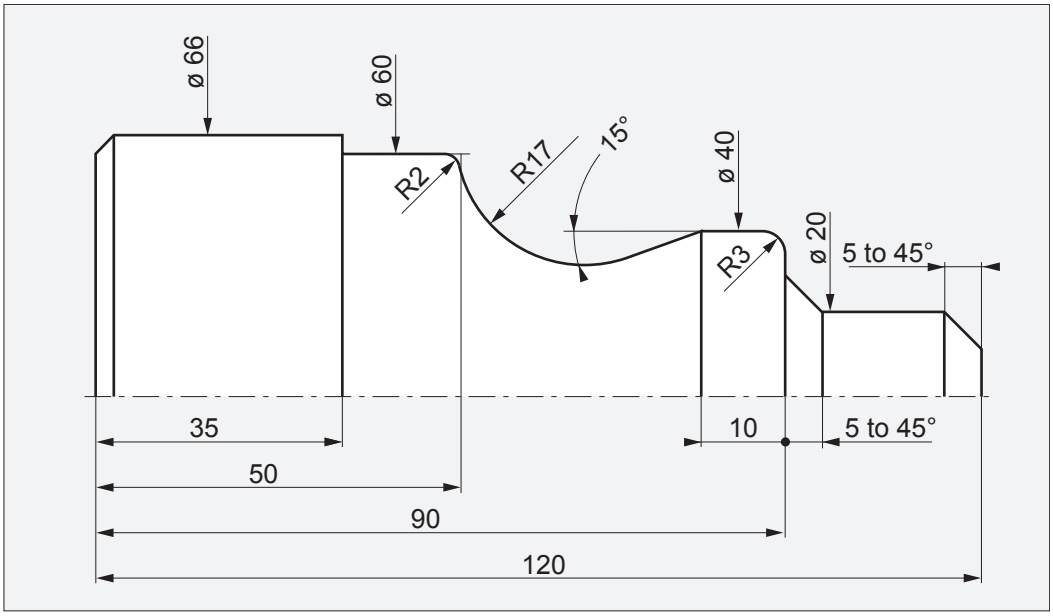
Machining paths



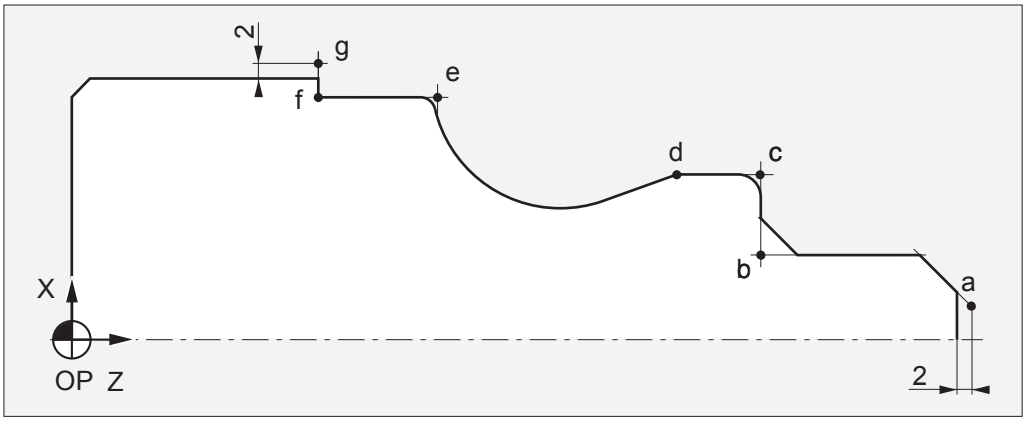

```
%18
N10 G00 G52 Z0
N20 T02 D02 M06 (CUTTER DIAMETER=6)
N30 S1200 M40 M03
N40 G92 R4                    Limiting of the tangential feed rate
N50 G00 G41 X61 Y20 Z3        Start point
N60 G01 Z-5 F50 M08           Penetration on Z
N70 Y10 F120
```

```
N80 G02 I55 J10
N90 G01 ET
N100 G03 I37 J-3 R7
N110 G01 EA-90 ES+
N120 G02 I0 J0 R50
N130 I-22.5 J-38.971
N140 G03 I0 J0 ES-
N150 G01 EA94
N160 G02 I0 J0 R15
N170 G01 EA79 Y40 EB10
N180 EA0 ES+
N190 G02 I4 J19.2 R32 ES-
N200 G01 EA180
N210 G03 I25 J21 R5
N220 G01 ET
N230 G02 I55 J10 X61 Y10
N240 G01 Y0 F500
N250 G00 G40 G52 Z0 M05 M09
N260 M02
```

Turned part #1 profile definition

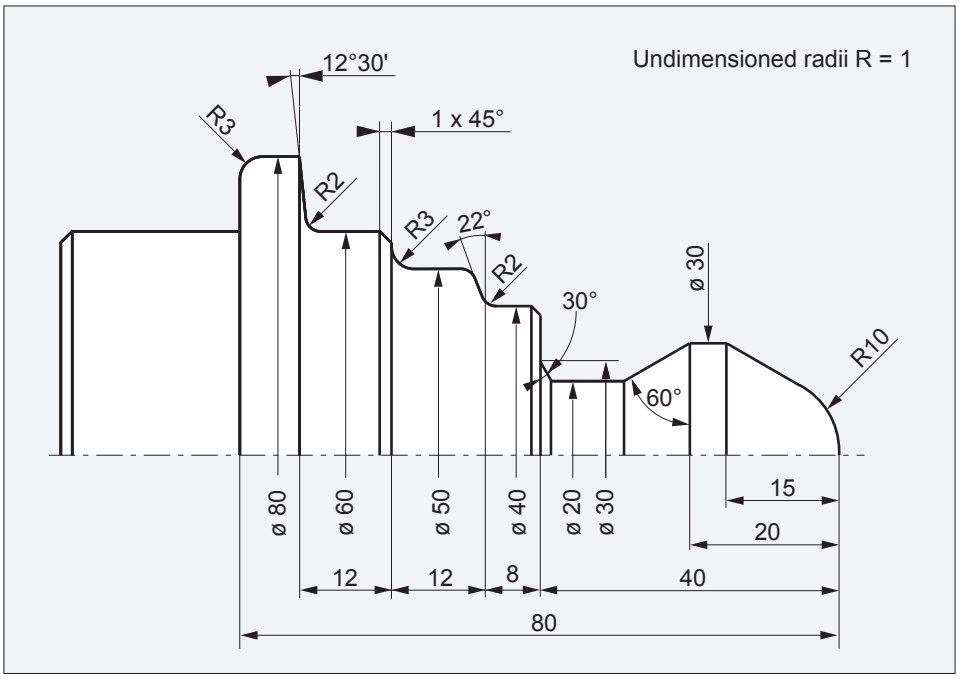


Machining paths

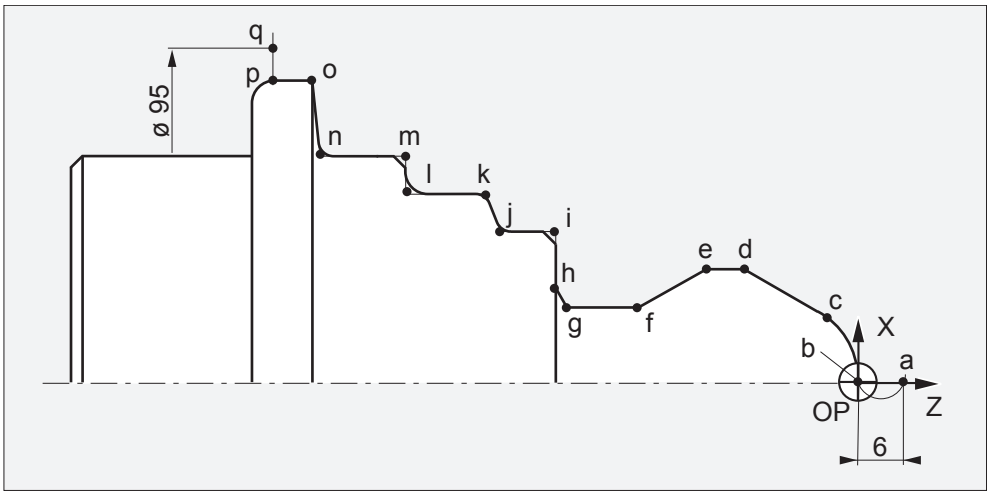



```
%88
N10 ...
N.. G92 S3000
N..
N90 G00 G52 X0 Z0
N100 T03 D03 M06 (COPYING TOOL R0.4)
N110 G97 S900 M04
N120 G00 G42 X6 Z122                    Point a, approach
N130 G96 S220
N140 G95 F0.1
N150 G01 EA135 ES
N160 EA180 X20 Z90 EB-5                 Point b
N170 X40 EB3                            Point c
N180 EA180 Z80 ES                       Point d
N190 EA195
N200 G02 X60 Z50 R17 EB2                Point e
N210 G01 Z35                            Point f
N220 X70                                Point g
N230 G00 G40 G52 X0 Z0 G97 S900
...
```

In PGP, part profile definition



Machining paths




```
%109
N10 ...
N.. G92 S3000
N..
N110 G00 G52 X0 Z0
N120 T05 D05 M06 (COPYING TOOL R0.4)
N130 G97 S900 M04
N140 G00 G42 X0 Z6                    Point a, approach
N150 G92 S3000
N160 G96 S150
N170 G95 F0.08
N180 G02 X0 Z0 I0 K3                  Point b
N190 G03 I0 K-10                      Point c
N200 G01 X30 Z-15                     Point d
N210 Z-20                             Point e
N220 EA-150 X20                       Point f
N230 EA180 ES                         Point g
```

```
N240 EA120 X30 Z-40                    Point h
N250 X40 EB1                           Point i
N260 EA180 Z-48 EB2                    Point j
N270 EA 112 X50 EB1                    Point k
N280 Z-60 EB3                          Point l
N290 X60 EB-1                          Point m
N300 EA180 ES EB2                      Point n
N310 EA102.5 X80 Z-72 EB1              Point o
N320 Z-78                              Point p
N330 G00 X95                           Point q
N340 G40 G52 X0 Z0 G97 S900
N..
```

Page deliberately empty

Page deliberately empty

Summary

7 Parametric Programming 455
7.1 "L" Local variables 455
7.1.1 Arithmetic Operations 456
7.1.2 Arithmetic Functions 456
7.1.3 Logic Operations 456
7.1.4 Comparison Symbols Used with L Variables 457
7.1.5 Conversion of the Internal Unit 457
7.2 Programming Syntax for L Variables 458
7.2.1 Assignment of a Variable to an NC function 458
7.2.2 Declaration of a Variable in the Program 459
7.2.3 Test of a Variable for a Conditional Branch 460
7.2.4 Notes on Programming Variables L100 to L199 and L900 to L959 461
7.2.5 Equivalence of Variables L900 to L925 461
7.2.6 Symbolic Addressing of Variables L900 to L925 and L926 to L951 461
7.3 Address Equivalences 470

Page deliberately empty

7 Parametric Programming

Parametric programming uses functions that can be assigned to all the NC addresses in place of numerical values and that can be used as particular functions.

Functions used in parametric programming:
- Program L variables,
- External E parameters.

7.1 "L" Local variables

Definition

Variables are elements that can be substituted for numerical values to provide a programming aid. Program variables are defined by letter address L followed by a number from one to three digits.

L variables represent decimal numbers are should be used in order to keep the maximum precision on floating point operation (divide, square root, trigonometric etc.)

List of L Variables

- Variables L0 to L19,
- Variables L100 to L199,
- Variables L900 to L959.

Variables L0 to L19, L100 to L199, L900 to L959 have the same format and use but cause differences in programming (see chapter 7.2).

Variable Assignments

L Variables can be assigned to all the programmable NC addresses.

The assignment of an L variable to an NC address automatically results in the correct use of the unit for the programmed address.

Initialisation

The variables are initialised to zero:
- at power on,
- at the end of the program (M02),
- after a reset.

Application

The values assigned to L variables can be:
- integer or decimal numbers (maximum 8 digits plus sign),
- fixed values or values resulting from operations.

Use

L Variables can be used:
- to perform operations,
- for increments and decrements,
- for conditional branches with function G79 after comparison with an expression, use caution with equivalence tests.
- jointly with programming of external E parameters to make transfers.

7.1.1 Arithmetic Operations

Arithmetic operation	Symbol	Arithmetic operation	Symbol
Addition	+	Multiplication	*
Subtraction	-	Division	/

Division by zero is impossible and results in error 94.

7.1.2 Arithmetic Functions

Arithmetic function	Symbol	Arithmetic function	Symbol
Sine	S	Arc tangent	A
Cosine	C	Square root	R
Truncation	T		

Sine (S) and cosine (C) The term following these functions is in degrees.

Truncation Extraction of the integer value of the number following the symbol.

Arc tangent The result of the operation is in thousandths of a degree.

Extraction of the square root of a negative number is impossible and results in error 94.

7.1.3 Logic Operations

Logic operation	Symbol	Logic operation	Symbol
AND	&	OR	!

AND and OR The operations are performed on values from which the decimal part is truncated (truncating performed automatically by the system) and which are expressed in binary.

7.1.4 Comparison Symbols Used with L Variables

Comparison symbol		Comparison symbol	
Equal to	=	Greater than or equal to	> =
Greater than	>	Less than or equal to	< =
Less than	<	Different from	< >

7.1.5 Conversion of the Internal Unit

In parametric expressions, functions U and M are used to convert values expressed in the internal system unit (see Chapters 2 and 3) to the programming unit:
- Function U is specific to linear axes
- Function M is specific to rotary axes.

Use of function U

Case of a system with micrometres (µm) as internal unit for linear axes.

Reminder: Programming in inches by G70 or metric system by G71.

- U254000 returns the value 254 for G71 (254 mm)
- U254000 returns the value 10 for G70 (10 inches).

Use of function M

Case of a system with 0.0001 degree as internal unit for rotary axes.

External parameter E74000 defines the reference position of axis 4 (B axis).

Conversion from 0.0001 degree (internal unit) to the programming unit in degrees. For any position on the B axis, the programming is:

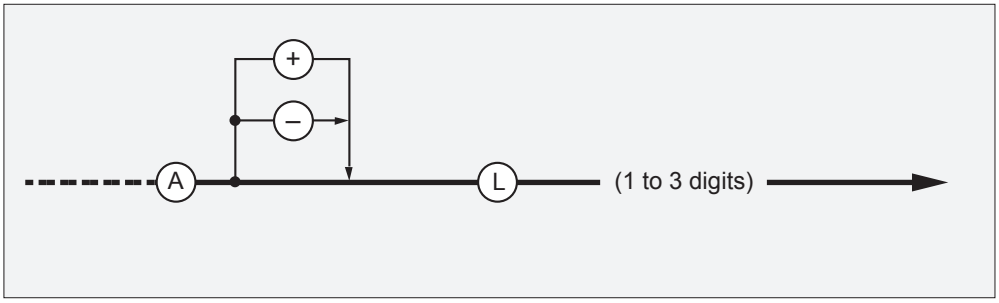
- L0=ME74000 or L0=[.IRX(8)]
- Variable L0 will hold the position on the B axis in degrees.

7.2 Programming Syntax for L Variables

Programming syntax for L variables is given as Conway diagrams and followed by programming examples.

7.2.1 Assignment of a Variable to an NC function

Syntax



A NC function.

+ / -Sign.

L Variable used as numerical value.

Example

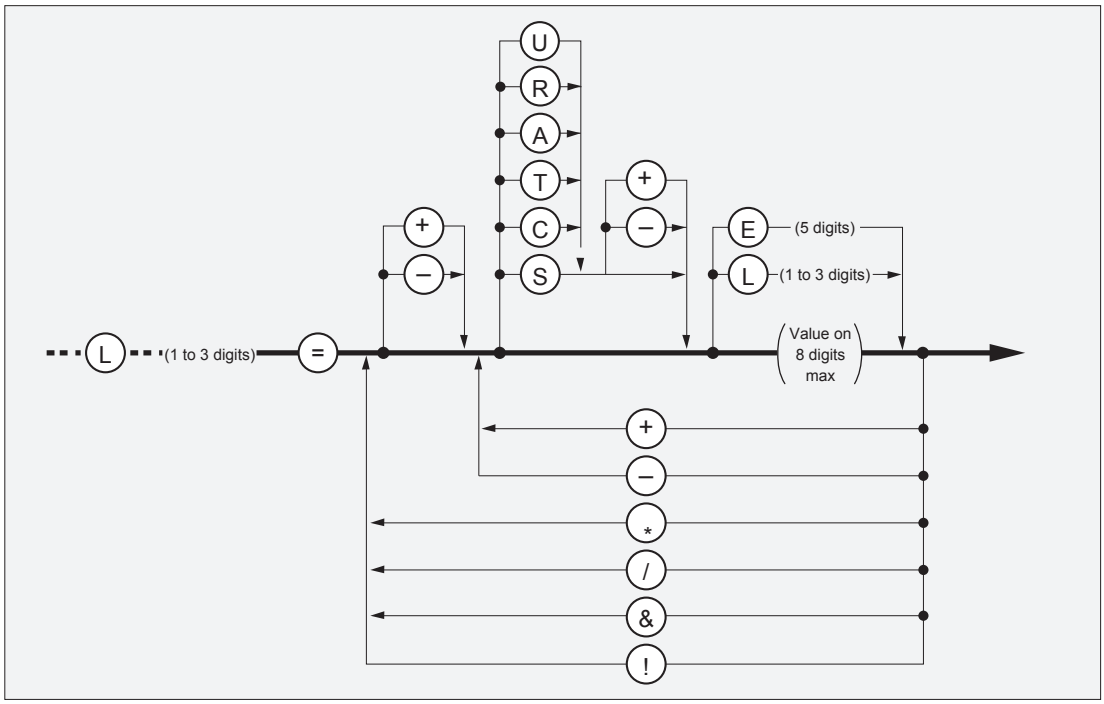
Use of a variable with NC addresses that have different units.

Assignment of variable L5 to addresses X and F.

```
N..
L5 = 18                          Declaration of the value of variable L5
N..
N.. G00 XL5                      L5 equals 18 mm with the X address
N..
N80 G94 G01 Z70 FL5              L5 equals a feed rate of 18 mm/min with the F address
N..
```

7.2.2 Declaration of a Variable in the Program

Syntax



The operations given in this diagram are detailed in the previous pages.

Notes

When the result of an operation giving a fractional number is assigned to an L variable, the system only keeps the first eight digits (integer and decimal parts) and ignores the rest. If the integer part of the result exceeds eight digits, the system reports an error.

Note on > = < Tests on L parameters

Be careful when making test on L variable. As an L variable is coded in Floating point the actual value can be slightly different from the displayed value. This is due to the principle of floating point calculation and cannot be considered as a default. The difference can be on the 10th or further digit after the decimal point. Therefore when making test on L values it is always better to check boundaries rather than exact value.

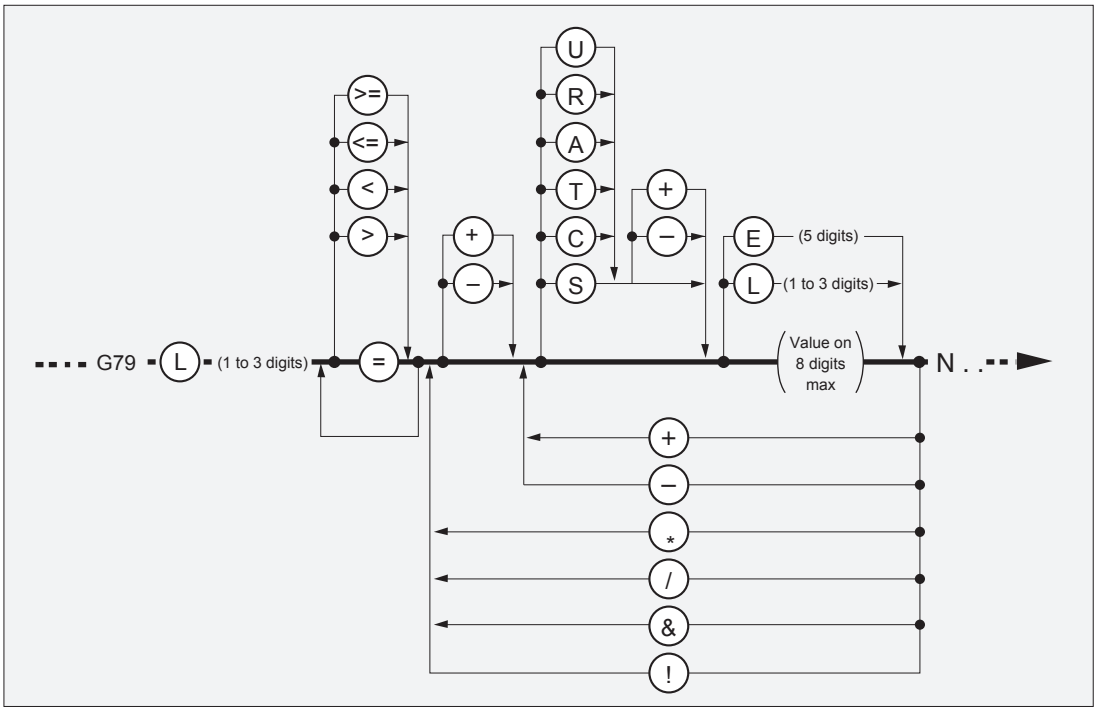
Example

Use of the variables with arithmetic operations.

`N..`	
`L1 = 5`	Declaration of the value of L1
`L2 = L1 + 5.3 * 3 * S30`	After the operation, L2 takes on the value 15.45 (sine 30° = 0.5)
`L3 = 100 / 3`	After the operation, L3 takes on the value 33.333333 (limitation to eight digits)
`N..`	
`N90 G00 XL2 Z30`	The value of L2 (15.45) is assigned to the X axis
`N100 XL3`	The value of L3 (truncated to 33.333 if the resolution is .001) is assigned to the X axis
`N..`	

7.2.3 Test of a Variable for a Conditional Branch

Syntax



Example

Use of a variable with a conditional test on the variable contents.

```
N..
N50 L1 = 0                          Initialisation of the variable with zero
N60 L1 = L1 + 1                     Increment of the variable
N..
N..
N200 G79 L1 < 10 N60                Condition: if L1 < 10, jump to N60; else continue
N210
N..
```

7.2.4 Notes on Programming Variables L100 to L199 and L900 to L959

Variables L100 to L199

Variables L100 to L199 have the same format and use as variables L0 to L19 but cause a different in programming.

Writing into a variable L100 to L199 suspends preparation of the block to which it belongs until execution of the previous block is completed. This is to prevent unexpected interaction if an external event (PLC generated subroutine or operator input in MDI for example) occurs that should generate a second block preparation sequence.

A block including a variable L100 to L199 cannot therefore be preceded by a block whose execution requires knowledge of the following block(s), e.g.:
- Profile Geometry Programming (PGP), or
- tool radius offset (G41, G42).

Programming of variables L100 to L199 with function M999

Writing of variables and transfer of current values into the part program are normally suspended until execution of the previous blocks is completed.

Programming M999 prohibits the operator or PLC from intervening in execution of a sequence of blocks, therefore allowing writing into these variables without stopping block preparation.

Variables L900 to L959

It is strongly not recommended to use variables L900 to L959 in a program with canned cycles (G81, G82, etc.), since they risk being overwritten when a cycle is called.

7.2.5 Equivalence of Variables L900 to L925

Variables L900 to L925 are equivalent to addresses A to Z respectively.

Example:

A = 250 is equivalent to L900 = 250

B = 1234 is equivalent to L901 = 1234 (and so forth to L925 for Z).

7.2.6 Symbolic Addressing of Variables L900 to L925 and L926 to L951

Variables L900 to L925 and L926 to L951 can be addressed by alphabetic symbols preceded by «'« (apostrophe). The variables can be used on the left or right side of an equation.

Variables L900 to L925

Variables L900 to L925 can be addressed by symbols 'A to 'Z respectively.

Example:

'C = 'A + 'B is equivalent to L902 = L900 + L901

Variables L926 to L951

Variables L926 to L951 can be addressed by symbols 'EA to 'EZ respectively ('EA = L926, 'EB = L927, etc. down to 'EZ = L951).

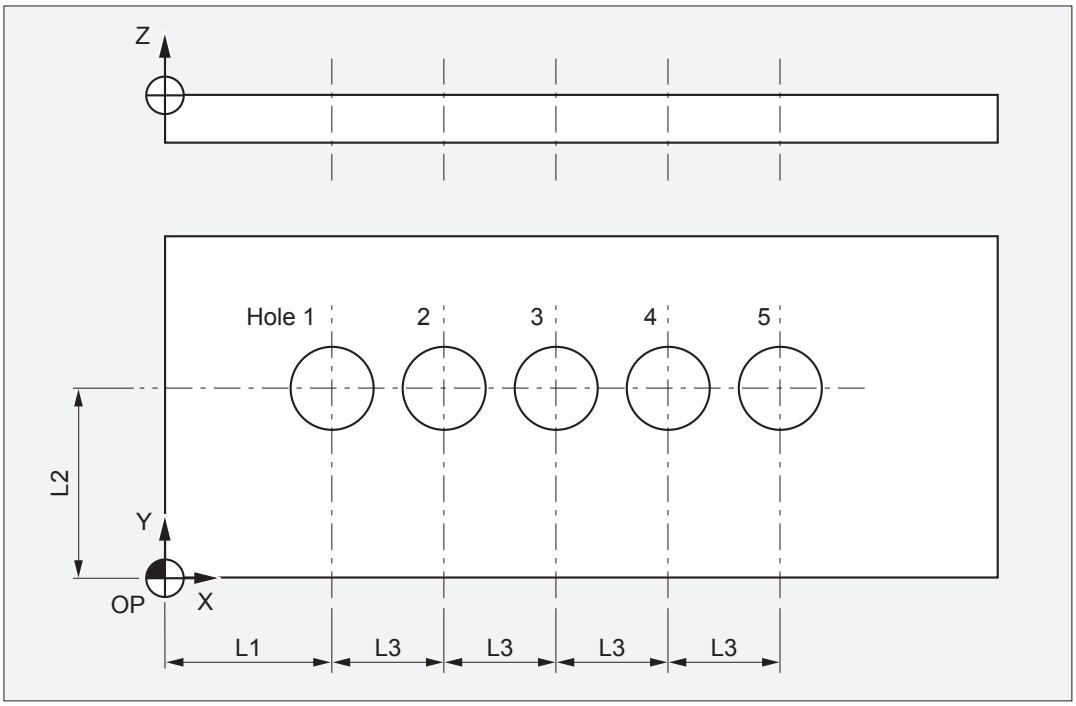
Example:

'A = 'B - 'EA/'EZ is equivalent to L900 = L901-L926/L951

Examples

Drilling holes with a conditional test

Execution of five holes in the XY plane (G17) with programming of a single drilling cycle.

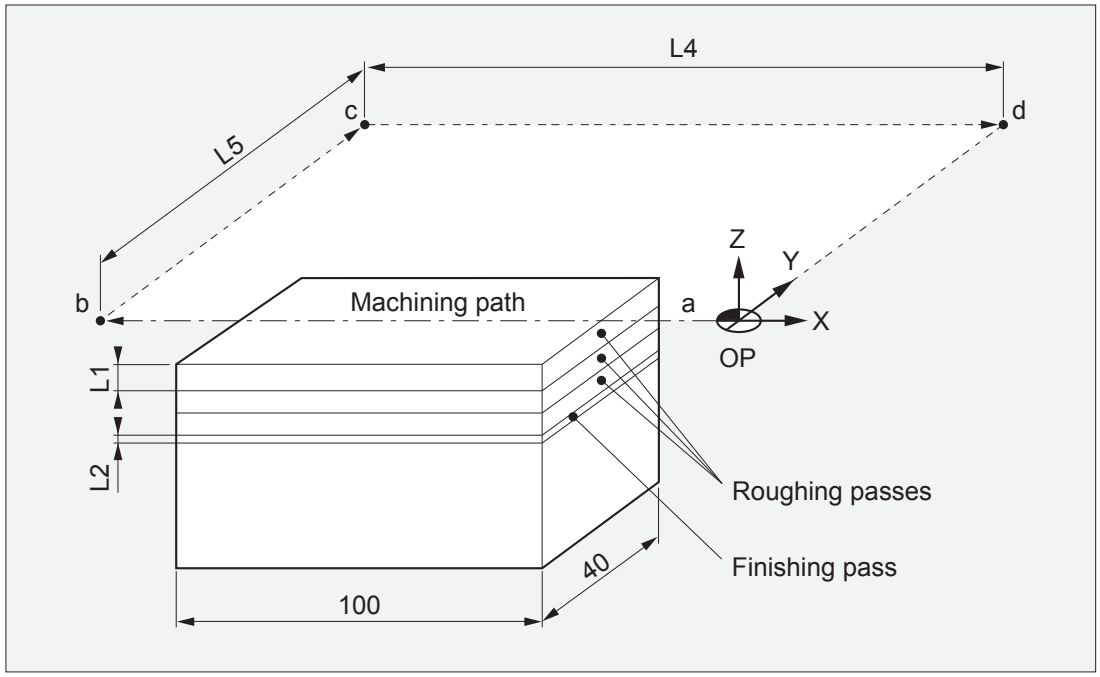



```
%40
L1=20 (START ON X)
L2=25 (START ON Y)
L3=15 (INCREMENT ON X BETWEEN HOLES)
N10 G00 G52 Z-50
N20 T01 D01 M06 (DRILL)
N30 S600 M40 M03
N40 XL1 YL2                    Positioning in the axis of hole 1
N50 L4=L4+1
N60 G81 Z-10 ER2 F100          Drilling in cycle
N70 G00 G80 G91 XL3            Movement in increment
N80 G90 G79 L4 < 5 N50         Condition: if L4 < 5, branch to block N50, else continue
N90 G00 G80 Z200 M05
N100 M02
```

Calculating the number of passes required

Facing in the XY plane (G17) by consecutive roughing and finishing passes in Z.
The diameter of the cutter is greater than the width of the part.

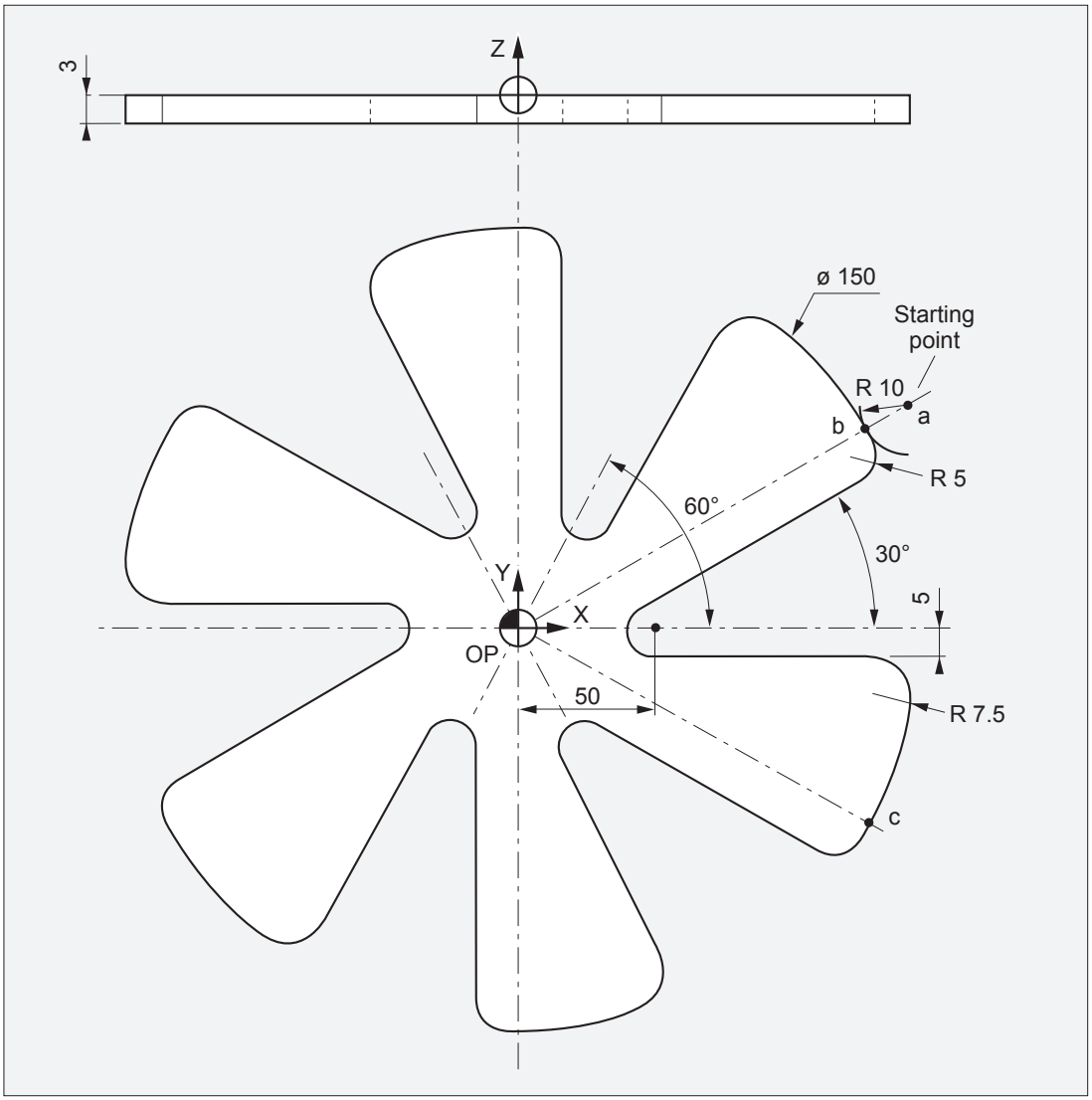
Program origin:
- X OP = Cutter radius plus clearance,
- Y OP = Cutter axis in the part axis,
- Z OP = Unmachined top surface of the part.




```
%25
$0 FACING Program
L1= 3 (Z VALUE OF ROUGHING PASS)
L2= 0.2 (Z VALUE OF FINISHING PASS)
L3= 3 (NUMBER OF ROUGHING PASSES)
L4= 200 (PASS LENGTH ON X)
L5= 100 (RETRACTION ON Y FOR RETURN)
L6= 150 (ROUGHING ROTATION SPEED)
L7= 120 (ROUGHING FEED RATE)
L8= 150 (FINISHING ROTATION SPEED)
L9= 80 (FINISHING FEED RATE)
N10 G00 G52 Z-50
N20 T01 D01 M06 (FACING CUTTER DIAMETER=50)
N30 SL6 M40 M03
N40 X0 Y0 Z30                          Point a
N50 Z0                                 Approach on Z0
N60 G91 Z-L1                           Pass setting on Z
N70 G01 X-L4 FL7 M08                   Point b
N80 G00 YL5                            Point c
N90 XL4                                Point d
N100 Y-L5                              Point e
N110 L3 = L3 - 1                       Decrement
N120 G79 L3 < 0 N140                   Condition: if L3 < 0, branch to block N40, else continue
N130 G79 N60
$0 FINISHING
N140 G00 Z-L2                          Finishing pass setting on Z
N150 G01 X-L4 SL8 FL9                  Finishing
N160 G00 G90 Z0 M05 M09
N170 M02
```

Calculating positions

Finishing of the profile of a gear wheel with use of PGP in the XY plane (G17)

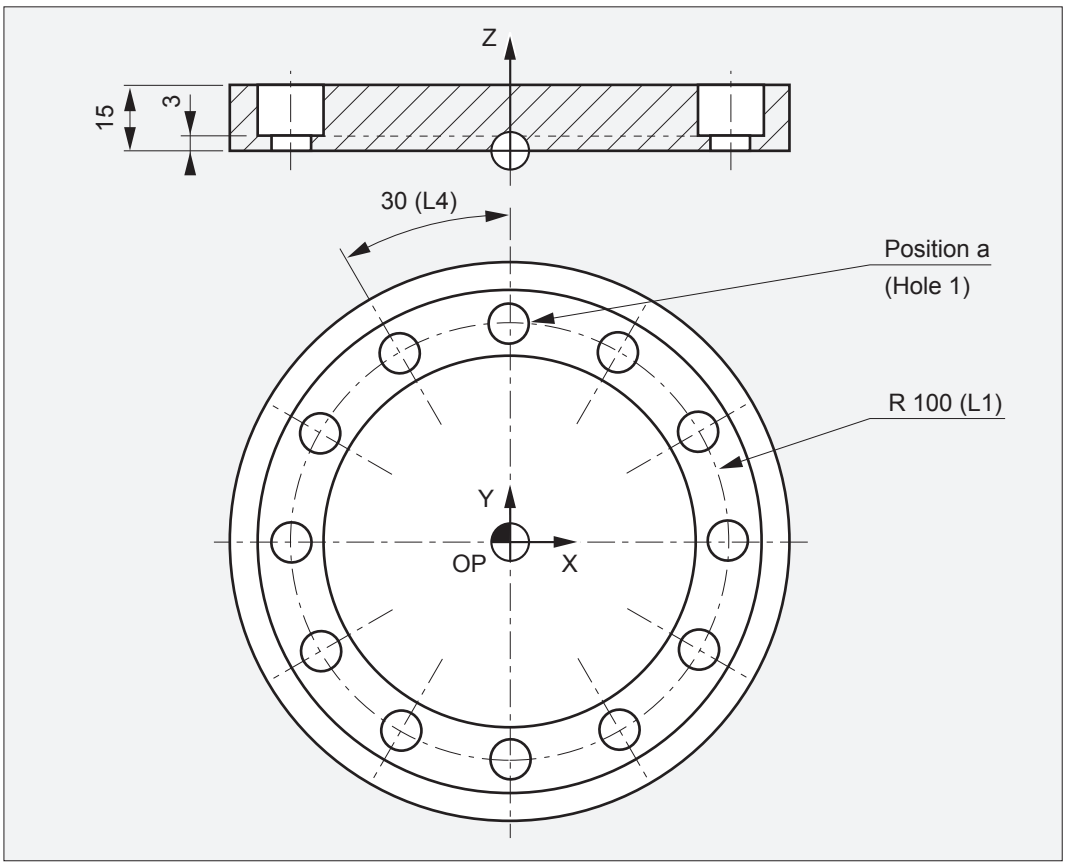



```
%21
N10 G00 G52 Z0
N20 T12 D12 M06 (CUTTER DIAMETER=8)
N30 S1000 M40 M03
L0 = 75 + 10 * C30                  Calculation of start point a on X
L1 = 75 + 10 * S30                  Calculation of start point a on Y
N35 G92 R1                          Limiting of tangential feed rate
N40 XL0 YL1 Z3 M08                  Point a, approach
N50 G01 Z-4 F50                     Approach on Z
N60 G41 EA180 ES+ F200
L3 = 75 * C30                       Calculation of point b on X
L4 = 75 * S30                       Calculation of point b on Y
N70 G03 XL3 YL4 IL0 JL1
N80 G02 I0 J0 ES- EB5
N90 G01 EA 210
N100 G03 I50 J0 R5
N110 G01 EA0 ES+ EB7.5
N120 G02 XL3 Y-L4 I0 J0             Point c
L10 = L10 - 60                      Decrement of the angular offset
```

```
N130 EDL10                          Angular offset
N140 G79 L10 > -360 N80             Test on 360°
N150 G03 IL0 JL1 R10 ES-
N160 G00 G40 EA90 XL0 YL1
N170 G40 G52 Z0 M05 M09
N180 M02
```

Drilling holes on a circle

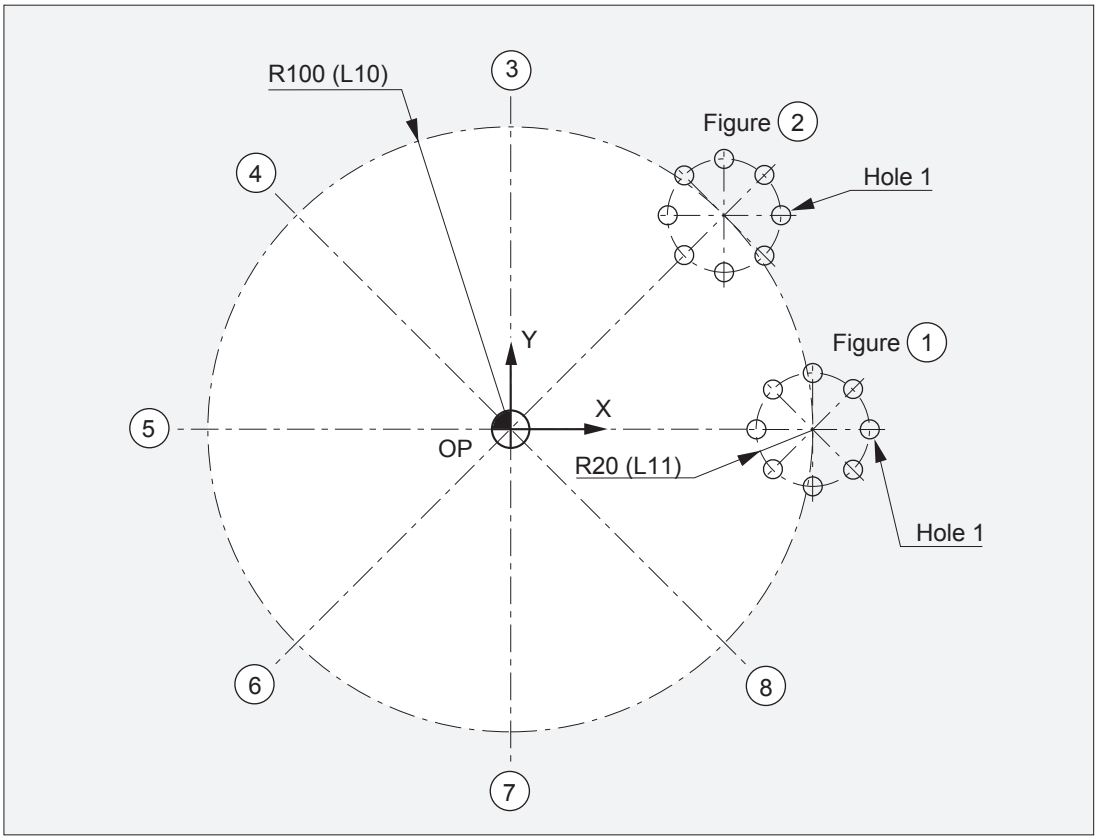
Movement by circular interpolation between holes.




```
%90
N10 G00 G52 Z0
N20 T01 D01 M06 (DRILL)
N30 S500 M03 M40
N40 L1=100 (RADIUS)
N50 L2=5 (APPROACH Z)
N55 L3=-5 (DRILLING Z)
N60 L4=30 (ED)
N65 L5=12 (NUMBER OF HOLES)
N70 X0 YL1                          Point a, approach on XY
N80 ZL2
N90 G79 N150                        Jump to block N150
N100 G81 ZL3 F100                   Cycle
N110 G80 G91 EDL4                   Angular offset by 30 degrees
N120 L10=L10+1                      Increment
N130 G79 L10=L5 N160                Condition: if L10 = 5, jump to block N160
N140 G90 G03 X0 YL1 I0 J0 F5000     Circular movement
N150 G77 N100 N140 SL5              Repeat the cycle
N160 G00 G52 Z-50
N170 G52 X-100 Y-50 M05
N180 M02
```

Calculating positions, Origin offsets and angular offsets

Drilling cycle with 8 circular patterns each including eight 6 mm diameter holes.



Main Program

```
%92
N10 G00 G52 Z0
N20 G52 X0 Y0
N30 T11 D11 M06 (DRILL DIA. 6)
N40 S4500 M03 M40
N50 L10=100 (RADIUS 1) L11=20 (RADIUS 2)
N60 L0=0 L1=L10*CL3 L2=L10*SL3 G59 XL1 YL2
N70 G77 H93                              Subroutine branch
N80 L3=L3+45 G79 L3<360 N60              Test on 360 degrees
N90 G00 G52 Z0 M05
N100 M02
```

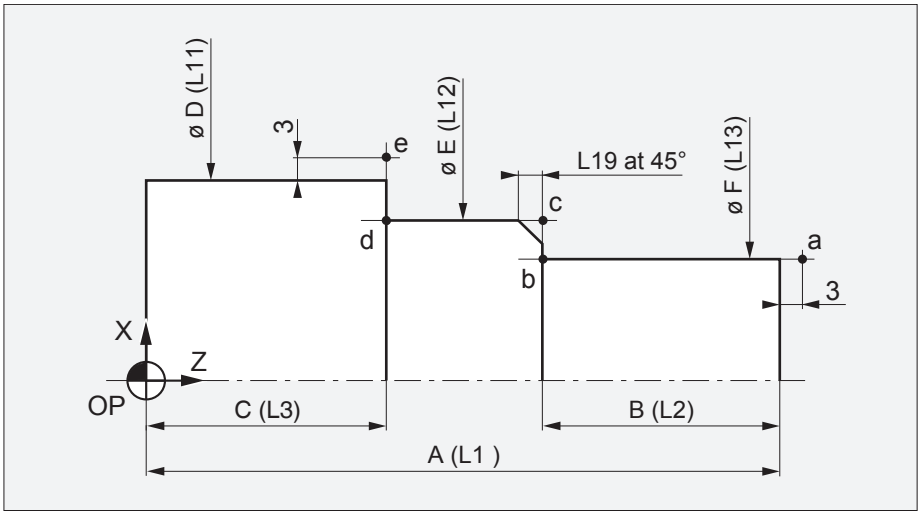
Subroutine

```
%93
N10 EDL0 G81 XL11 Y0 ER2 Z-10 F500       Cycle
N20 L0=L0+45 G79 L0<360 N10              Test on 360 degrees
N30 G80 G00 Z5
```

Examples

Programming L Variables

Use of variables for machining parts of a family
Machining of part 2 according the table.



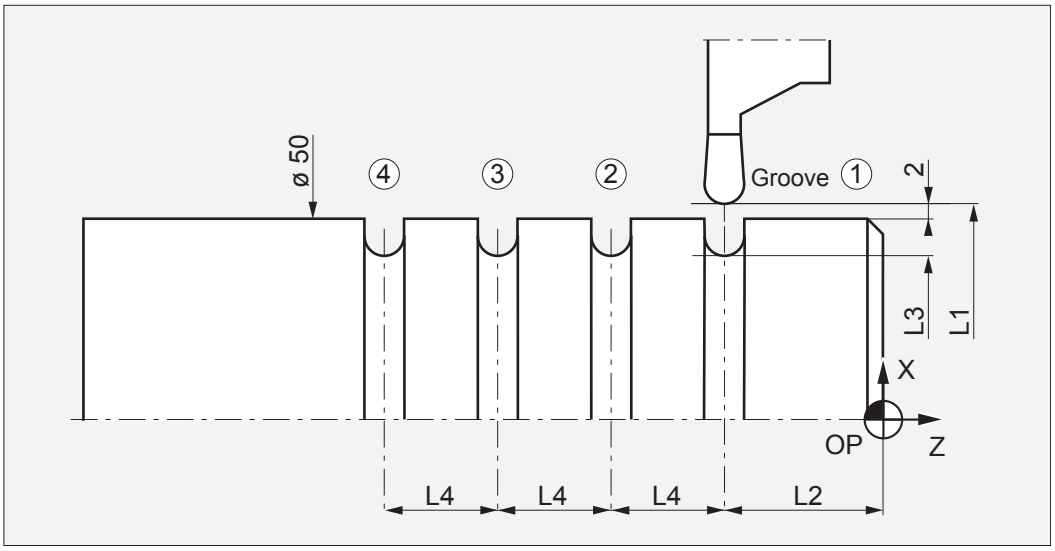
Variable assignment

Variables	L1	L2	L3	L11	L12	L13	L19
Dimension	A	B	C	D	E	F	Chamf.
Part 1	80	30	30	50	40	30	2
Part 2	85	28	35	48	39	28	1.5

```
%18
L1=85 (LENGTH A)
L2=28 (LENGTH B)
L3=35 (LENGTH C)
L11=48 (DIAMETER D)
L12=39 (DIAMETER E)
L13=28 (DIAMETER F)
L19=1.5 (CHAMFER)
N10 G00 G52 X0 Z0
N20 T01 D01 M06 (EXTERNAL TOOL)
N30 G97 S900 M40 M03
N40 G95 F0.2
L4=L1+3                          Computation of point Za
N50 G42 XL13 ZL4                 Point a
N60 G92 S2000
N70 G96 S160
L5=L1-L2                         Computation of point Zb
N80 G01 ZL5                      Point b
N90 XL12 EB-L19                  Point c
N100 ZL3                         Point d
L6=3*2+L11                       Computation of point Xe
N110 XL6                         Point e
N120 G40 G77 N10 N10 G97 S900 MO5
N130 M02
```

Use of variables with conditional test on the number of grooves

Execution of four grooves by programming a single G83 cycle.




```
%42
L1=54 (START X)
L2=20 (START Z)
L3=40 (GROOVE BOTTOM DIAMETER)
L4=15 (STEP BETWEEN GROOVES)
L10=50 ((CONSTANT SURFACE SPEED) L11=0.1 (FEED RATE))
N10 G00 G52 X0 Z0
N20 G16 P+
N30 T01 D01 M06 (GROOVING TOOL)
N40 G97 S900 M40 M03
N50 XL1 Z-L2                    Groove 1 axis position
N60 L5=L5+1                     Increment a counter
N70 G92 S2000
N80 G96 SL10
N90 G83 XL3 P4 Q1 G95 FL11      Groove in cycle
N100 G00 G80 G79 L5=4 N130      Condition: if L5=4 go
                                to block N130
N110 G91 Z-L4                   Spacing between two grooves
N120 G90 G79 N60                Branch to execute the next groove
N130 G77 N10 N10 G97 S900 M05
N140 M02
```

7.3 Address Equivalences

This table is used to specify the equivalence between two axes. For instance, it allows a program to be written for another channel, such as UW.

The symbol @ followed by an address (A to Z) specifies an equivalent axis.

The declaration of an equivalent address is programmed by:

@ [A " " Z] = [A " " Z]

The assignment of a value to an equivalent function is programmed by:

@ [A " " Z] followed by the value

Example :

```
@ X = U
```

@ X/300 is equal to U 300

The address equivalence table is reset at power on, by a reset or by M02 (@A = A, @B = B, ..., @Z = Z).

The declaration of a new equivalent address (@ [A " " Z] = [A " " Z]) suspends preparation of the block in which it is programmed until execution of the previous block is completed.

Page deliberately empty

Page deliberately empty

Summary

8 External E Parameters 475
8.1 E Parameters - Main summary table 475
8.2 E Parameters description 480
8.2.1 Arithmetic Operations 481
8.2.2 Arithmetic Functions 481
8.2.3 Logic Operations 481
8.2.4 Comparison Symbols Used with External E Parameters 482
8.2.5 Conversion of the Internal to Program Unit 482
8.2.5.1 Examples 482
8.3 External E Parameter Programming Syntax 483
8.3.1 Assignment of an External Parameter to an NC Function 483
8.3.2 Declaration of an External Parameter in the Program 484
8.3.3 Test on an External Parameter for a Conditional Branch 485
8.4 How to read an E parameters table description 486
8.5 List of External E Parameters 488
8.5.1 Parameter for exchange with the machine application (PLC) 488
8.5.2 Program Analysis Access Parameters 491
8.5.3 NC Configuration access parameters 498
8.5.4 Input / Outputs 501
8.5.5 Machining Parameters 503
8.5.6 Channel Axis Access Parameters 509
8.5.7 Spindle Access Parameters 516
8.5.8 Common NC Parameters 519
8.5.9 Machine Axis Access Parameters 521
8.5.10 Axis adjustment 527
8.5.11 Special purpose parameters 529
8.5.11.1 Delayed power down on machine errors 529
8.5.11.2 Axis / Spindle switching parameters 530
8.5.11.3 Spindle synchronisation 531
8.5.11.4 Drive access parameters 533

Page deliberately empty

8 External E Parameters

8.1 E Parameters - Main summary table

Click on **Exxxx** to give the detailed information.

E parameter	Description
E10000 to E10031	Boolean I/O parameters for exchange with the PLC application
E110xx	Special machining cycles
E11000	Angular offset (function ED..) enabled
E11001	Program origin offset (function G59) enabled
E11003	Processing of mirror function (function G51) enabled
E11005	Programming by diameter enabled
E11006	Tool center programming (Turning Machine)
E11007	Spindle speed potentiometer enabled
E11008	Drawing of a Complete Circle
E11009	Complete revolution in G1 on rotary axes
E11010	Shortest rotation direction selection
E11012	Zero following error operation
E11013	Gradual acceleration
E11014	Deceleration over several blocks
E11015	Angle crossing management enabled
E11016	Selects front or rear turret (Turning Machine)
E11017	Inclined plane function enabled (G24+)
E11018	RTCP function enabled (G26+)
E11019	Storing axes position on hardware interrupt
E11020	Allows writing the reference position for a "virtual" axis
E11021	Activation of the speed averaging filte outside HSC
E11022	Emergency retract cause
E20000 to E20031	Boolean input parameters for exchange with the PLC application
E20100 and E20101	Status of the probing (interrupt) inputs
E21000 to E21255	Presence of Software options bits 0 to 255
E30000 to E30255	32 bits I/O parameters for exchange with the PLC application
E31002	Part program zone protection
E32000	Minimum interpolation block execution time
E32001	Overspeed coefficien on the path in G12
E32002	Servo-control error tolerated on a circle
E32003	Minimal angle for speed analysis
E32004	Chordal error
E32005	Number of total speed anticipation filter terms
E32006	Time to reach maximum acceleration
E32008	Time to reach a constant jerk (derivative of the acceleration)
E32009	Feed rate limitation in case of curve radius variation
E32010	Coefficien (in 1/256) of the maximum radial acceleration
E32011	Number of terms of the speed averaging filter
E32012	Error tolerated on circles programming

E parameter	Description
E33000 to E33999	Addressing of the PLC Outputs
E34000 and E34001	8 bits parameters to access the NCK embedded digital outputs
E34100 and E34101	16 bits parameters to access the NCK embedded analogues outputs
E34200 and E34201	Digital Output Locked by PLC
E34300 and E34301	Analog Output Locked by PLC
E35100 to E35131	First 16 bits test point value
E35132 to E35163	Second 16 bits test point value
E35164 to E35195	32 bits test point value
E35200 to E35231	Code number of the first test point
E35232 to E35263	Code number of the second test point
E35264 to E35295	Enable the reading test point
E353xx	Select a set of drive parameters
E36000 to E36999	16 bits parameters to access the PLC memory
E37000 to E37127	8 bits parameters to access the PLC
E40000 to E40255	32 bits input parameters to access the PLC memory
E41000 to E41006 and E41102	Parameters returning encoded value
E41000	Current CNC mode
E41001	Current channel number
E41002	Number of channels
E41004	System Serial number
E41005	Sampling period
E41006	Channel position servo-loop time constant
E41010	Interrupt channel triggered
E41102	Number of NC channels
E43000 to E43999	Addressing of the PLC Inputs
E44000 and E44001	8 bits parameters to access the NCK embedded digital inputs
E44100 to E44103	16 bits parameters to access the NCK embedded analogues inputs
E46000 to E46999	16 bits parameters to access the PLC memory
E47000 to E47127	8 bits parameters to access the PLC memory
E49001 to E49128	Dynamic Operation number
E50000	Current tool correction number
E50xxx	Tool length L of correction xxx or X dimension for Turning Machines
E51000	Current tool axis orientation
E51xxx	Cutter tip radius @ of correction xxx or Z dimension for Turning Machines
E52xxx	Tool radius R of correction xxx
E53xxx	Dynamic correction DL or DX for tool offset xxx
E54xxx	Dynamic radius correction DR (or DZ) for tool offset xxx
E55xxx	Tool tip orientation for turning tools
E56xxx	Available parameters (H of the table of dynamic corrections)
E57xxx	Parameter defining the tool type xxx
E58xxx	Tools geometry code

E parameter	Description
E6x000	DAT1 on axis x (x = axis address 0 to 8)
E6x001	DAT2 on axis x (x = axis address 0 to 8)
E6x002	Minimum dynamic machine travel on the axis "x"
E6x003	Maximum dynamic machine travel on the axis "x"
E6x004	Eccentricity offsets on axis "x" (DAT3)
E6x005	Offset programmed by G59 on axis "x"
E69000	Scaling factor
E69001	Inclined axis angle (grinders)
E69002	Inclined wheel angle (grinders)
E69003	Axis re-assignment
E69004	Tool disengagement in G41/G42 for an angle bigger that 120°
E7x000	Position reference of an axis in the channel
E7x001	Sampling the axes positions for on-the-fly probing
E7x002	Minimum static machine travel
E7x003	Maximum static machine travel
E7x004	Direction of movement of the axes being interpolated
E7x005	Axis address assignment
E7x006	Axis coupling
E7x007	Axes programmed by diameter
E7x100	Interpolator position references
E7x101	Interpolation speed limiting
E79000	Reference position of the currently measured spindle
E79001	Speed demand for the currently controlled spindle
E79002	Value of the feed rate potentiometer assigned to the channel
E79003	Distance remaining to be travelled in the block being executed
E79004	Current feed rate on the path programmed in the block
E79005	Minimal value for feed override
E79006	Maximum value for feed override
E79007	Minimal value for spindle override
E79008	Maximum value for spindle override
E79009	Distance to go within the block
E79010	Distance remaining in the current programmed block
E79011	Expansion factor
E8x000 to E8x999	Parameters related to the inter-axis calibration
E80000 to E80199	Local 32 bits parameters
E81000 to E81999	Master axis position reference
E82000 to E82999	Slave axis corrections with respect to the master axes
E83000 to E83999	General purpose 32 bits parameters
E84000 to E84999	General purpose 32 bits parameters
E9yyxx	Parameters for physical address of the axis (0 to 31)
E900xx	Axis measurement
E9010x	Spindle x position reference
E9011x	Spindle x modulo
E9020x	Spindle x speed setting
E9030x	Maximum spindle x indexing speed
E9031x	Spindle x in-position window

E parameter	Description
E9032x	Spindle x indexing gain
E9033x	Spindle x indexing acceleration
E9034x	Speed threshold below which spindle x is considered stopped
E9035x	Tolerance interval coefficien
E90120 to E90123	C axis / Spindle measurement conversion used for Spindle/ axis switching
E90224 to E90227	Spindle synchronisation errors
E90360 to E90363	Enable synchronisation on master spindle
E90370 to E90373	Spindle rotation speed for maximum torque
E90380 to E90383	Spindle rotation speed for maximum power
E910xx	Servo or non-servo status of the axes
E911xx	Homing status on an axis
E912xx	N/M AUTO axes
E913xx	Clampable axis enable state
E920xx	Axis origin switch enable state
E92100 to E92131	Axis error masking
E922xx	Scaling the lag display
E923xx	Enable or disable Phase lag and Phase lead compensation
E92400 to E92403	Spindle error masking
E930xx	Axis @xx home switch status
E931xx	Measured axis
E932xx	Axis declared as rotary modulo 360 degrees
E933xx	Axis homing direction
E934xx	Homing status without switch wiring
E935xx	Axis (or spindle) in-position window
E93700 to E93731	Indication of axis in error or not
E93800 to E93803	Indication of spindle in error or not
E940xx	Parameters related to inter-axis calibration
E941xx	Association of a slave axis (or spindle) with a master axis (or spindle)
E94124 to E94127	Master spindle number for spindle synchronisation
E942xx	Association between drive and axis or spindle command
E943xx	Physical handwheel coupler in the list of handwheels
E950xx	Axis reference correction
E951xx	Position of the origin switch with respect to the machine origin
E952xx	Axis (or spindle) measurement correction
E953xx	Differential offset value
E960xx	Duplicated axis in automatic modes
E961xx	Duplicated axis in manual mode (JOG)
E962xx	Synchronised axis
E963xx	Symmetrically driven axis
E970xx	Maximum axis speed
E971xx	Axis acceleration at programmed feed rate
E972xx	Axis acceleration in Rapid
E973xx	Maximum authorised speed at angle crossings
E98024 to E98027	Spindle integral action time constant
E981xx	Axis acceleration anticipation time constant
E982xx	Amplitude of the antistick pulse on reversal

E parameter	Description
E983xx	Time constant to absorb the antistick pulse
E984xx	Delay to apply the drive reference
E98500 to E98563	Partial feed forward per axis
E98600 to E98663	Anticipation time constant (firs 32 values) and Lag balancing error filter time constant (last 32 values)

8.2 E Parameters description

Definition

External E parameters are used by the part program to access information contained in the PLC or CNC memory.

External parameters are defined by address letter *E* followed by five digits.

Application

Depending on the external E parameters, the part program accesses the CNC or PLC memory for *Read only or Read/Write* operations.

The value assigned to an external E parameter is always an integer.

Assignment

The assignment of an external parameter to an NC address causes the same units to be used for the E parameter and the programmed address.

Initialisation

The parameters E100xx, E200xx, E30xxx, E33xxx, E34xxx, E36xxx, E37xxx, E40xxx, E43xxx, E44xxx, E46xxx, E47xxx, E41xxx, E41001, E41002, E83000 and E84000 are initialised on power-up. The other external parameters are never reset by the system.

Use of the External E Parameters

The external parameters can be used:
- for operations,
- for increments and decrements,
- for conditional branches with function G79 after comparison with an expression,
- jointly with L variables to make transfers.

Precautions for Use

The use of the external parameters obeys certain rules:
- parameters E20000, E40000, E41000, E41001, E41002, E70000 cannot be *«written»* by the program as they are read-only parameters,
- an operation on an external parameter stops movement at the end of the previous block,
- a block including an external parameter cannot be preceded by a block whose execution requires knowledge of the following block(s).

  Example
  - Profile Geometry Programming (PGP),
  - Tool radius offset (G41, G42).

8.2.1 Arithmetic Operations

Arithmetic operation	Symbol	Arithmetic operation	Symbol
Addition	+	Multiplication	*
Subtraction	-	Division	/

Division by zero results in error 94.

8.2.2 Arithmetic Functions

Arithmetic function	Symbol	Arithmetic function	Symbol
Sine	S	Arc tangent	A
Cosine	C	Square root	R

Sine (S) and cosine (C) The term following these functions is in degrees.
Truncation Extraction of the integer value of the number following the symbol.
Arc tangent The result of the operation is in degree.

Extraction of the square root of a negative number results in error 94.

8.2.3 Logic Operations

Logic operation	Symbol	Logic operation	Symbol
AND	&	OR	!

AND and OR The operations are performed on binary values.

8.2.4 Comparison Symbols Used with External E Parameters

Comparison symbol		Comparison symbol	
Equal to	=	Greater than or equal to	> =
Greater than	>	Less than or equal to	< =
Less than	<	Different from	< >

8.2.5 Conversion of the Internal to Program Unit

Conversion	Symbol
Linear axes	U
Rotary axes	M

In parametric expressions, functions U and M are used to convert values expressed in the internal system unit (see Chapters 2 and 3) to the programming unit:
- Function U is specific to linear axes
- Function M is specific to rotary axes.

8.2.5.1 Examples

Case of a system with micrometres (µm) as internal unit for linear axes

- U254000 returns the value 254 for G71 (254 mm)
- U254000 returns the value 10 for G70 (10 inches).

Case of a system with 0.0001 degree as internal unit for rotary axes

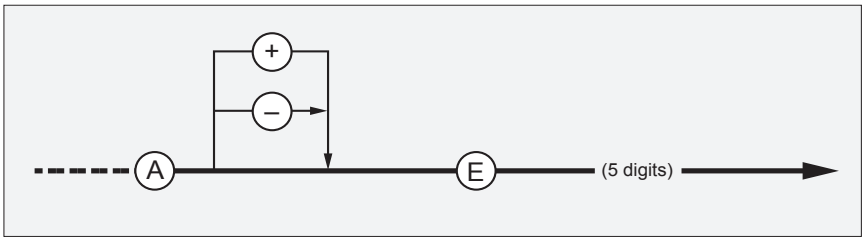
Conversion from 0.0001 degree (internal unit) to the programming unit in degrees.
- L0=ME74000 returns the B axis position in degrees.

8.3 External E Parameter Programming Syntax

The programming syntax for external E parameters is given as Conway diagrams followed by programming examples.

8.3.1 Assignment of an External Parameter to an NC Function

Syntax



A NC function.

+ / - Sign.

E External parameter used as numerical value.

Notes

External E parameters have integer values. When they are assigned to a function with a decimal value, the decimal point is implicit and depends on the format of the function.

When the value of the parameter is not compatible with the format of the function (too many digits, etc.), the system reports an error.

Example

Use of an external parameter with NC addresses in different units.

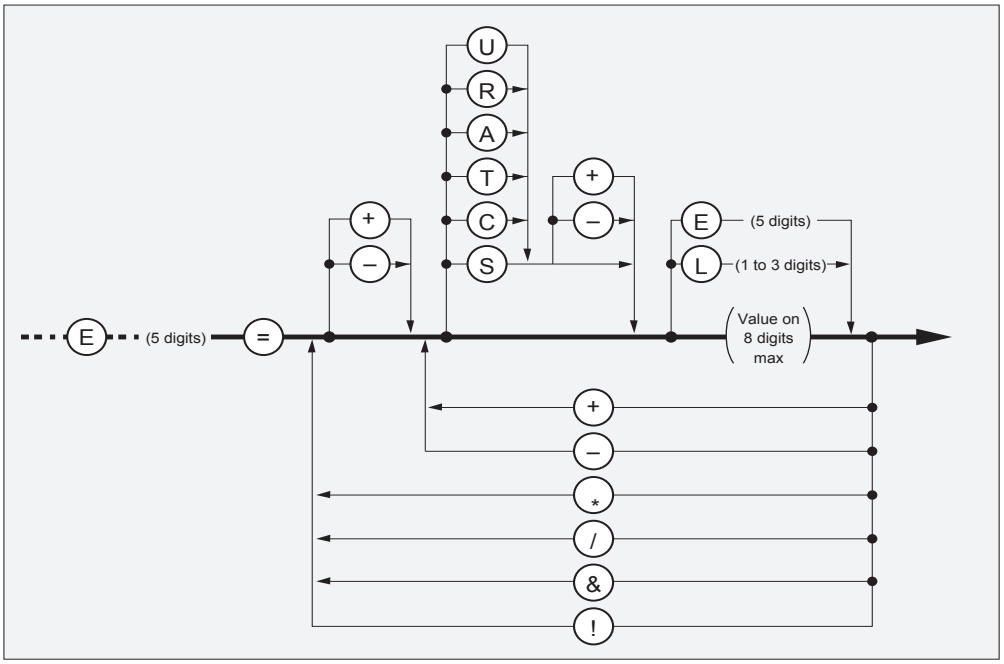
Assignment of external parameter E80000 to addresses X and B (C for Turning).

```
N..
E80000 = 18000          Declaration of the value of the parameter
G00 XE80000             The value of E80000 is equivalent to 18 millimetres
                        (internal unit = µm)
G00 BE80000             The value of E80000 is equivalent to 1.8 degrees
                        (format B034, i.e.  ten-thousandths of a degree)
```

It should be noted that if the feed rate F is programmed as FE80000 (example), the feed rate F takes on the integer value declared in parameter E (in this case, the decimal digits are excluded with F).

8.3.2 Declaration of an External Parameter in the Program

Syntax



The operations given in this diagram are detailed in the previous pages.

Notes

When an operation result that is not an integer is assigned to an external parameter, the decimal digits are truncated. To keep a result with decimal digits, this result must be assigned to a variable L (see chapter 7.2).

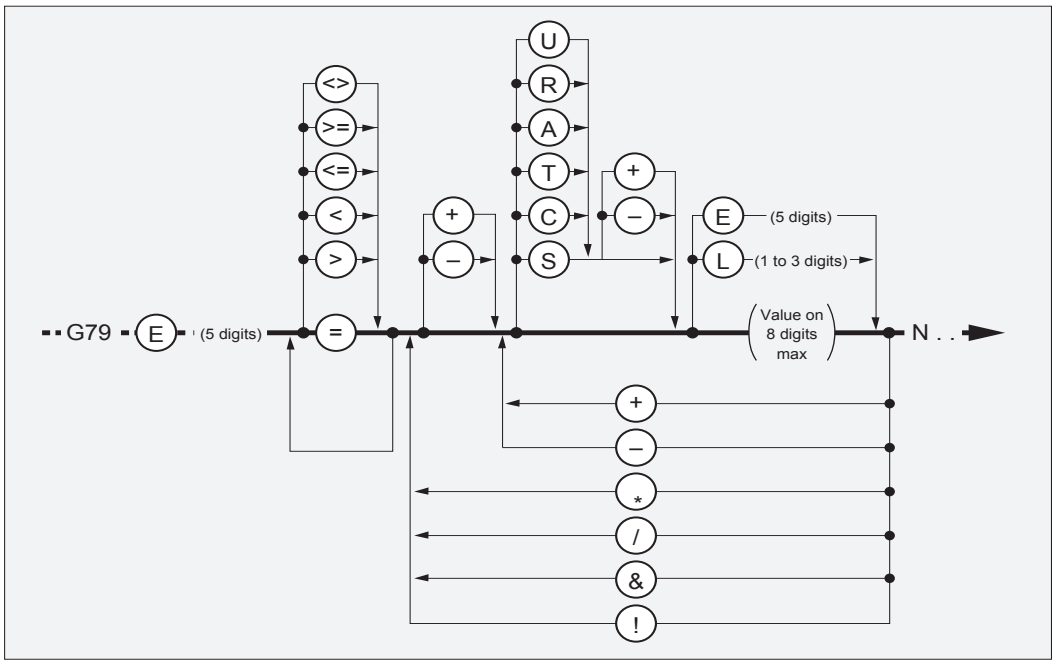
Example

Use of external parameters with arithmetic operations.

```
N..
E80002 = 3150                    Declaration of the value of E80002
E80016 = 2400                    Declaration of the value of E80016
E80005 = E80002 / E80016         After the operation, E80005 takes on the value 1
                                 (truncation)
L1 = E80002 / E80016             After the operation, L1 takes on the value 1.3125
N..
```

8.3.3 Test on an External Parameter for a Conditional Branch

Syntax



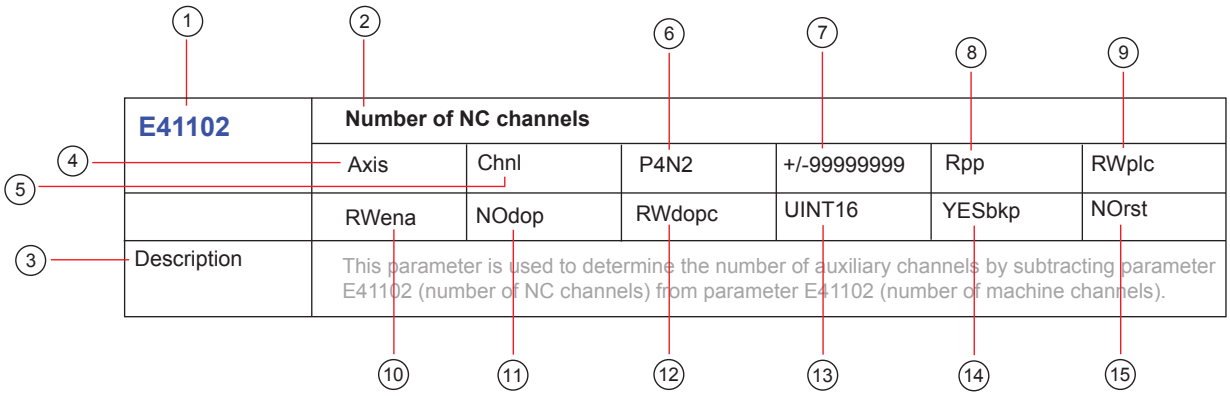
Example

Use of an external parameter with a conditional test.

```
N.. ...
N50 E56003 = E56003+1
G79 E56003 = > 10 N50
N..
```

Increment of the parameter
Condition: if E56003 ≥10, branch to N50, else continue

8.4 How to read an E parameters table description



1	**E parameter number**	
2	**E parameter label**	
3	**E parameter description**	
4	**Related function**	**All**: Available for all functions/application
		Prg: Part Programming
		Tools: Tool data
		Spindle: Available with spindle application
		Axes: Available with axes application
		HSC: High Speed Cutting
		Drive: Available with Digital Drive
		I/O: Input / Output for PLC
5	**Channel specific**	**Chnl**: Channel specific parameter
		Cmn: Parameter common for all channels
6	**Unit**	**None**: Unscaled number
		P4N2: Linear unit measure depending from P4N2 configuration
		P4N4: Angular unit measure depending from P4N4 configuration
		P4 N2..N4: Linear or angular unit measure according to the axis configuration
		ms: Time expressed in milliseconds
		... : Other International System units
7	**Admitted value**	**+/-99999999:** Integer value from -99999999 to +99999999
		0/1: Only 0 or 1 integer values
		00/01/10/11: Only 0 or 1 or 10 or 11 integer values
		0/1/-1: Only 0 or 1 or -1 integer values
		.....
8	**Access from Part Program**	**Rpp**: Read only
		Wpp: Write only
		RWpp: Read/Write
		NOpp: No access
9	**Access from PLC variables**	**Rplc**: Read only
		Wplc: Write only
		RWplc: Read/Write
		NOplc: No access

10	**Access from PLC ENA**	**Rena**: Read only
		Wena: Write only
		RWena: Read/Write
		NOena: No access
11	**Access from Dynamic Operator** Note. Optionnaly the specification can be followed by a number in parentheses (n). This means only the 'n' first items get this property. Exemple: E33000 "Wdop (10)"... We can Write with Dynamic operators "BUT ONLY THE FIRST 10".	**Rdop**: Read only
		Wdop: Write only
		RWdop: Read/Write
		NOdop: No access
12	**Access from Dynamic Operator in C** See note as for position 11 above.	**Rdopc**: Read only
		Wdopc: Write only
		RWdopc: Read/Write
		NOdopc: No access
13	**C programm dimension**	**UINT8**: integer 8 bit ($0 \div 2^8 -1$)
		UINT16: integer 16 bit ($0 \div 2^{16} -1$)
		UINT32: integer 32 bit ($0 \div 2^{32} -1$)
		SINT32: integer 32 bit ($-2^{31} \div +2^{31} -1$)
		float: $-2^{127} \div +2^{127}$
14	**Backup**	**YESbkp**: Saved on power off
		NObkp: Not saved on power off. Restarted with 0 value.
15	**Affected by NCK reset (RAZ)**	**NOrst**: Not affected, the value doesn't change after reset
		P…N… : On NCK reset the E parameter assume the value of the NCK parameter P word N

8.5 List of External E Parameters

8.5.1 Parameter for exchange with the machine application (PLC)

E10000 to E10031	Boolean I/O parameters for exchange with the PLC application					
	All	Cmn	None	0/1	RWpp	Rplc
	Rena	RWdop	Rdopc	UINT32	NObkp	NOrst
Description	Parameters used to report a boolean information to the PLC program on "Flexium_NCK.RCNC.E10000[x]" variable. The parameters are sent without waiting for a report by the PLC. They can be read or written by part program and dynamic operators. If a value different from 0, positive or negative, is input to the E parameter, the NC doesn't report any error and manages it as "1". The function of these parameters is defined by the machine builder. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E20000 to E20031	Boolean input parameters for exchange with the PLC application					
	All	Cmn	None	0/1	Rpp	Wplc
	Rena	Rdop	Wdopc	UINT32	NObkp	NOrst
Description	Parameters used to retrieve a boolean information from the PLC program on "Flexium_NCK.WCNC.E20000[x]" variable. They can be read by part program and dynamic operators. The function of these parameters is defined by the machine builder. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E30000 to E30255	32 bits I/O parameters for exchange with the PLC application					
	All	Cmn	None	+\-999...	RWpp	Rplc
	Rena	NOdop	NOdopc	...	NObkp	NOrst
Description	Parameters used to report significant signed numerical values to the PLC program on "Flexium_NCK.RCNC.E30000[x]" variable. The parameters are sent without waiting for a report by the PLC. They can be read or written by part program. The function of these parameters is defined by the machine builder.					

E33000 to E33999	Addressing of the PLC Outputs					
	I\O	Cmn	None	0...255	Wpp	Rplc
	NOena	Wdop (10)	Wdopc	UINT8	NObkp	NOrst
Description	Parameters used to report to the PLC Output image via dedicated "Flexium_NCK.RCNC.E33000[x]" variables (999 unsigned integer [8 bits]). The parameters are sent without waiting for a report by the PLC. They can be written by part program. If a value out from the range 0 ... +255 is entered to the E parameter, the NC doesn't report any error and manages its less significant byte The first 10 parameters (up to E33009) can be written via dynamic operator. The function of these parameters is defined by the machine builder. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E36000 to E36999	16 bits I/O parameters to access the PLC memory					
	I\O	Cmn	None	0...65535	RWpp	Rplc
	Rena	Wdop (3)	Wdopc	UINT16	NObkp	NOrst
Description	Parameters used to report numerical values to the PLC program on "Flexium_NCK.RCNC.E36000[x]" variable. The parameters are sent without waiting for a report by the PLC. They can be read and written by part program. If a value out from the range 0 ... +65535 is entered to the E parameter, the NC doesn't report any error and manages its less significant word. The first 3 parameters (up to E36002) can be written via dynamic operator. The function of these parameters is defined by the machine builder. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E37000 to E37127	8 bits output parameters to access the PLC memory					
	All	Cmn	None	0...255	NOpp	Rplc
	Rena	Wdop	RWdopc	UINT8	NObkp	NOrst
Description	Parameters used to report numerical values to the PLC program on "Flexium_NCK.RCNC.E37000[x]" variable. The parameters are sent without waiting for a report by the PLC. They can be written via dynamic operator only. If a value out from the range 0 ... +255 is entered to the E parameter, the NC doesn't report any error and manages its less significant byte. The function of these parameters is defined by the machine builder. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E40000 to E40255	32 bits input parameters to access the PLC memory					
	All	Cmn	None	0... 2^{32} -1	Rpp	Wplc
	Rena	NOdop	NOdopc	...	NObkp	NOrst
Description	Parameters used to retrieve significant signed numerical values from the PLC program on "Flexium_NCK.RCNC.E40000[x]" variable. They can be read by part program. The function of these parameters is defined by the machine builder.					

E43000 to E43999	Addressing of the PLC Inputs					
	I\O	Cmn	None	0...255	Rpp	Wplc
	NOena	Rdop (10)	Rdopc (10)	UINT8	NObkp	NOrst
Description	Parameters used to retrieve from the PLC Input image via dedicated "Flexium_NCK.RCNC.E43000[x]" variables.They can be read by part program. The first 10 parameters (up to E43009) can be read via dynamic operator. The function of these parameters is defined by the machine builder. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E46000 to E46999	16 bits input parameters to access the PLC memory					
	All	Cmn	None	0...65535	Rpp	Wplc
	NOena	Rdop (3)	Rdopc (3)	UINT16	NObkp	NOrst
Description	Parameters used to retrieve numerical values from the PLC program on "Flexium_NCK.RCNC.E46000[x]" variable. They can be read by part program. The first 3 parameters (up to E46002) can be read via dynamic operator. The function of these parameters is defined by the machine builder. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E47000 to E47127	8 bits input parameters to access the PLC memory					
	All	Cmn	None	0...255	Rpp	Wplc
	Rena	Rdop	Rdopc	UINT8	NObkp	NOrst
Description	Parameters used to retrieve numerical values from the PLC program on "Flexium_NCK.RCNC.E47000[x]" variable. They can be read by part program and via dynamic operator. The function of these parameters is defined by the machine builder. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

8.5.2 Program Analysis Access Parameters

These parameters are generally set by the machine builder via custom made macros. Do not attempt to modify these values except if you are familiar with precise servo adjustment as the results on the part being machined can be drastic especially in HSC.

E110xx	The use of these parameters is reserved for the creation of special machining cycles. Unless otherwise specified, these parameters are of the read/write type and are used to enable or inhibit geometric transformations and speed overrides. They have values of 0 or 1 and belong to a channel. A «0» inhibits transformation and a «1» enables transformation, i.e. it is carried out if the function using it is programmed or if it is enabled by default. Unless otherwise specified, these parameters are set to 1 by a reset.

E11000	**Angular offset (function ED..) enabled**					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	Parameter used to enable or disable the angular offset function (ED).					

E11001	**Program origin offset (function G59) enabled**					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	Parameter used to enable or disable taking into account the programmed origin offsets. Please also refer to the G59 on chapter 5.					

E11003	**Processing of mirror function (function G51) enabled**					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	Parameters used to enable or disable the mirror function. Please also refer to the G51 on chapter 5.					

E11005	**Programming by diameter enabled**					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	This parameter is reset to force programming by radius on the X (or U) axis. Programming by diameter is possible only with G20 (programming in X, Y, Z polar coordinates) provided it is enabled by machine parameter P4. Please also refer to the G20 on chapter 5 and P4 on Flexium Parameters manual M00021.					

E11006	Tool center programming (Turning Machine)					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	Setting this parameter is equivalent to a tool orientation C0. The main use is generally for graphic simulation.					

E11007	Spindle speed potentiometer enabled					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	When this parameter is set to «0» for a channel, the spindle speed used for this channel is set to 100 percent.					

E11008	Drawing of a Complete Circle					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	With circular interpolation, this parameter is set to «0» to prevent a complete circle when the start point and end point are the same. This parameter is set to "1" after a reset, i.e. permitting full circles.					

E11009	Complete revolution in G1 on rotary axes					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	This parameter when set to 0 prevents the execution of a complete turn when the same position is programmed for a rotary axis. It is set to 0 after a reset therefore permitting full revolution in one block. Remark. In G0 if the same position is programmed, no movement occurs whatever the value in E11009. Please also refer to the G0 and G01 on chapter 5.					

E11010	Shortest rotation direction selection					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	This parameter when set to 0 select the shortest path for a rotary axis (except if a sign is explicitly programmed). It is set to "1" after a reset therefore the direction of rotation is selected by the sign preceding the positioning value.					

E11012	Zero following error operation					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	0
Description	For high precision contouring, this parameter enables cancellation of the following error. The parameter is set to enable cancellation and reset to inhibit it. A system reset does not affect the value of parameter E11012. Please refer to machine parameter P19 on Flexium Parameters manual M00021.					

E11013	Gradual acceleration					
	Prg	Chnl	None	-1\0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	P7N1.5
Description	This parameter is set to enable S ramp accelerations. It is reset to vary the speed linearly. At power on, this parameter is set as declared in machine parameter P7 (bit 5, word 1). Among the E110xx parameter, this is the only one with three possible values: 0 : constant acceleration 1 : Linear acceleration -1 : Trapezoidal acceleration. Please refer to machine parameter P7 on Flexium Parameters manual M00021.					

E11014	Deceleration over several blocks					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	The deceleration function is enabled when this parameter is set to «1» and disabled when it is set to «0». A reset sets this parameter to «1».					

E11015	Angle crossing management enabled					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	1
Description	When this parameter is set to «0», the angles are not analysed when calculating the end of block speed.					

E11016	Selects front or rear turret (Turning Machine)					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	P5N1.0
Description	Parameter used on lathe to check if the turret is a front turret (1) or a rear turret (0).					

E11017	Inclined plane function enabled (G24+)					
	Prg	Chnl	None	0\1	Rpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	0
Description	This parameter set to indicate that the inclined plane function (G24+) is enabled. Please also refer to the G24+ on chapter 5.					

E11018	RTCP function enabled (G26+)					
	Prg	Chnl	None	0\1	Rpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	0
Description	This parameter set to indicate that the RTCP function (G26+) is enabled. Please also refer to the G26+ on chapter 5.					

E11019	Storing axes position on hardware interrupt					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	0
Description	When E11019 is set the system will record the axes position on each interrupt like in G10 except that the interruption will not terminate the current block. Please also refer to the G10 on chapter 5.					

E11020	Allows writing the reference position for a "Virtual" axis					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	0
Description	This parameter is used for allowing to write the position on a 'Virtual' axis (P2 = 0, P3 = 1). For more detailed information on "Virtual" axis please refer to Flexium Extended manual M00020					

E11021	Activation of the speed averaging filter outside HSC					
	Prg	Chnl	None	0\1	RWpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	0
Description	This parameter is used to activate a speed averaging system on systems. It excludes using HSC simultaneously. If this speed averaging filter is used, then HSC is not available.					

E11022	Emergency retract cause					
	All	Cmn	None	0 / 1	Rpp	NoPlc
	Rena	Nodop	Nodopc	UNIT32	Nobkp	0
Description	This parameter is set to 1 when an emergency retract is triggered automatically by lost of communication between NC and PLC (see P99 N3) . It will be reset at the next cycle start.					

E31002	Part program zone protection					
	Prg	Cmn	None	0\1	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	0
Description	When set this parameter prevents any change in the part program area. This feature is useful when binary files are accessed by dynamic operators through their physical adresses. A relocation of these files is therefore forbiden leaving the adresses valid.					

E32000	Minimum interpolation block execution time					
	Prg	Chnl	ms	0..65535	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	P51N0
Description	Parameter defining the minimum execution time for a block containing a linear or circular interpolation, expressed in milliseconds (ms). This parameter can be modified by programming. It only affects the channel in which it is programmed (it is initialised with the value of P51 by a reset). Please refer to machine parameter P51 on Flexium Parameters manual M00021.					

E32001	Overspeed coefficient on the path in G12					
	Prg	Cmn	1\1024	0..32767	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	NOrst
Description	Parameter defining the handwheel overspeed coefficient expressed in 1/1024 (with G12). The variations of the increments on the first handwheel multiplied by this coefficient generate an overspeed applied to the programmed path. The new value written is saved until the NC is reset. Please also refer to the G12 on chapter 5.					

E32002	Servo-control error tolerated on a circle					
	Prg	Cmn	P4N2	0..32767	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	P52N0
Description	Parameter defining the servo-control error tolerated on a circle, expressed in NC internal measurement unit(P4N2). It is the image of machine parameter P52. Please refer to machine parameter P52 on Flexium Parameters manual M00021.					

E32003	Minimal angle for speed analysis					
	HSC	Cmn	Degrees	0..360	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	0
Description	Parameter defining the angle above which angle crossing speed is always analysed. This parameter is set to «0» at power on. Its value **in degrees** must be declared by programming or this processing is always carried out. The last value programmed is saved after a reset.					

E32004	Chordal error					
	HSC	Cmn	P4N2	0..65535	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	10 / NOrst
Description	Parameter defining the chordal error in Internal linear units. At power on, this value is initialised with 10 and can be modified by programming. The last value programmed is saved after a reset.					

E32005	Number of total speed anticipation filter terms					
	HSC	Chnl	None	0..13	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	P55N8...N15 / NOrst
Description	Parameter defining the number of terms of the total speed anticipation filter. This parameter is assigned to all the axes of the channel in which it is declared. Write of this parameter is enabled if: - the «high precision contouring» option is present; otherwise the system returns error message 4, - anticipation is enabled (E11012 = 0), or the system returns error message 95, - it is below 14, or the system returns error message 94. The last value programmed is saved after a reset. Please refer to machine parameter P55 on Flexium Parameters manual M00021.					

E32006	Time to reach maximum acceleration					
	HSC	Cmn	ms	0..65535	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	P53N0
Description	This parameter defines the time to get the maximal acceleration. It is used with the controlled jerk feature. The default value is defined by machine parameter P53 N0. Please refer to machine parameter P53 on Flexium Parameters manual M00021.					

E32008	Time to reach a constant jerk (derivative of the acceleration)					
	HSC	Cmn	ms	0..65535	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	0 / NOrst
Description	When non zero, it allows for trapezoidal jerk. Set to 0 at power up but maintained after a reset. Please refer to machine parameter P53 on Flexium Parameters manual M00021.					

E32009	Feed rate limitation in case of curve radius variation					
	HSC	Cmn	None	0..65535	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	0
Description	This parameter is used in HSC to define the error tolerated in case of curvature variation. The value is defined as 1500 * square root of the tolerated error.					

E32010	Coefficient (in 1/256) of the maximum radial acceleration					
	HSC	Cmn	1\256	0..65535	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	256
Description	Range from 0 to 65535 with zero equivalent to 256.					

E32011	Number of terms of the speed averaging filter					
	RTCP	Cmn	None	0..13	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	0 / NOrst
Description	To be used outside HSC. Number of terms of the speed averaging filter.					

E32012	Error tolerated on circles programming					
	RTCP	Cmn	P4N2	0..255	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	P52N2
Description	This parameter defines the biggest difference between the start radius and the end radius in circular interpolation triggering error 107. This parameters represents the machine parameter P52N2 and is reinitialised with its value after a reset (base 20 increments). Please refer to machine parameter P52 on Flexium Parameters manual M00021.					

E49001 to E49128	Dynamic Operation number					
	Prg	Cmn	None	0..25	Rpp	NOplc
	Rena	NOdop	NOdopc	...	NObkp	0
Description	These parameters giving the dynamic operator numbers declared in operations 1 to 128. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

8.5.3 NC Configuration access parameters

These parameters are generally used to create custom macros for specific function. Do not attempt to modify them without a good knowledge of the application installed on the machine.

E20100 and E20101	Status of the probing (interrupt) inputs					
	Prg	Cmn	None	0/1	Rpp	NoPlc
	NOEna	NODop	NODopc	...	Nobkp	Norst
Description	These two parameters give the status of each of the probing signals located on the X9 NCK connector. E20100 corresponds to PB0 on pin 13 and E20101 to PB1 on pin 15. The parameters are set to 1 if the corresponding input signal is at 24Vdc, they are equal to 0 otherwise.					

E21000 to E21255	Presence of Software options bits 0 to 255					
	Prg	Cmn	None	0\1	Rpp	NOplc
	Rena	NOdop	Rdopc	UINT32	NObkp	NOrst
Description	Parameters addressing binary values. These parameters test for presence of s/w options 0 to 255. When parameter is set the option is present.					

E41000 to E41006 and E41102	Parameters returning encoded value
Description	These parameters are generally used to conditionally execute or ignore some part program sections.

E41000	Current CNC mode					
	All	Chnl	None	0..15	Rpp	NOplc
	Rena	NOdop	Rdopc	UINT8	NObkp	NOrst
Description	This parameter holds a number corresponding to the current mode: - 0: continuous mode - 1: single mode - 2: MDI mode - 3: rapid mode - 4: sequence number search mode - 6: test mode - 7: manual mode - 8: home mode.					

E41001	Current channel number					
	All	Chnl	None	0...7	Rpp	NOplc
	Rena	NOdop	Rdopc	UINT8	NObkp	NOrst
Description	This parameter read in a program returns the channel number to which the program is assigned. (channel 1: E41001=0, channel 2: E41001=1, etc.).					

E41002	Number of channels					
	All	Cmn	None	1...8	Rpp	NOplc
	Rena	NOdop	Rdopc	UINT8	NObkp	NOrst
Description	This parameter returns the number of channels available in the system.					

E41004	System Serial number					
	All	Cmn	None	0..6xxxxx	Rpp	NOplc
	Rena	NOdop	Rdopc	UINT32	NObkp	NOrst
Description	This parameter returns the serial number of the CNC.					

E41005	Sampling period					
	All	Cmn	µs	0...2^{32}-1	Rpp	NOplc
	Rena	NOdop	Rdopc	UINT32	NObkp	P50
Description	This parameter returns the axis sampling period in microseconds.					

E41006	Channel position servo-loop time constant					
	All	Chnl	ms	0..1200	Rpp	NOplc
	Rena	NOdop	Rdopc	UINT32	NObkp	NOrst
Description	This parameter defines the time constant in milliseconds (ms). It is the image of machine parameter P56. Please refer to machine parameter P56 on Flexium Parameters manual M00021.					

E41010	Interrupt channel triggered					
	All	Cmn	None	0 ..3	Rpp	NoPlc
	Rena	Nodop	Rdopc	UINT8	Nobkp	0
Description	This flag indicates on which interrupt channel an interrupt has occured. Values E41010 = 1 when PB_EV1 = 1 E41010 = 2 when PB_EV2 = 1 E41010 = 0 in case of reset.					

E41102	Number of NC channels					
	All	Cmn	None	1...8	Rpp	NOplc
	Rena	NOdop	Rdopc	UINT8	NObkp	NOrst
Description	This parameter is used to determine the number of auxiliary channels by subtracting parameter E41102 (number of NC channels) from parameter E41002 (number of channels).					

8.5.4 Input / Outputs

The role of the following parameters is defined by the machine builder. For additional information, please refer to the machine instruction manual.

E34000 and E34001	8 bits parameters to access the NCK embedded digital outputs					
	I\O	Cmn	None	0..255	RWpp	RWplc
	Rena	Wdop	RWdopc	UINT8	NObkp	NOrst
Description	Parameters used to access the NCK embedded digital outputs: E34000 = out7..out0 connector X10. E34001 = out15..out8 connector X10. They can be read or written by part program and only written via dynamic operators. If a value out from the range 0 ... +255 is entered to the E parameter, the NC doesn't report any error and manages it as modulo 256 of the absolute value. If a value out from the range 0 ... +255 is entered to the E parameter, the NC doesn't report any error and manages its less significant byte. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators" Reminder. The NCK embedded digital outputs can be managed via Parameters E3400x if the PLC variable Flexium_NCK.WCNC.IO.LockDigitalOutput[x]:= FALSE The status of these PLC flags is also available for reading in parameters E34200 and E34201.					

E34100 and E34101	16 bits parameters to access the NCK embedded analogues outputs					
	I\O	Cmn	None	0..65535	RWpp	RWplc
	Rena	Wdop	RWdopc	UINT16	NObkp	NOrst
Description	Parameters used to access the NCK embedded analogue outputs: E34100 = Analogue out0 connector X9. E34101 = Analogue out1 connector X9. They can be read or written by part program and only written via dynamic operators. The relationship between value and output voltage is: 0 -> 0V 32767 -> +10V 32768 -> 0V 65535 -> -10V or in other word 0...32767 -> 0..10V with the bit 15 of the value used to define the voltage sign. A second opportunity is offered only in case of value assignment: E... = -3267 is like E... = 36044 -> -1V If a value out from the range 0 ... +65535 is entered to the E parameter, the NC doesn't report any error and manages its less significant word. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators" Reminder: The NCK analogue outputs can be managed via E3410x parameters if the PLC variable Flexium_NCK.WCNC.IO.LockAnalogOutput[x]:= FALSE The status of these PLC flags is also available for reading in parameters E34300 and E34301.					

E34200 and E34201	Digital Output Locked by PLC					
	I/O	Cmn	None	0/1	Rpp	NOplc
	NOEna	NOdop	NOdopc	UINT8	NObkp	NOrst
Description	These two parameters give the status of "Flexium_NCK.WCNC.IO.LockDigitalOutput[x]" PLC variables.					

E34300 and E34301	Analog Output Locked by PLC					
	I/O	Cmn	None	0/1	Rpp	NOplc
	NOEna	NOdop	NOdopc	UINT8	NObkp	NOrst
Description	These two parameters give the status of "Flexium_NCK.WCNC.IO.LockAnalogOutput[x]" PLC variables.					

E44000 and E44001	8 bits parameters to access the NCK embedded digital inputs					
	I\O	Cmn	None	0..255	Rpp	Rplc
	Rena	Rdop	Rdopc	UINT8	NObkp	NOrst
Description	Parameters used to access the NCK embedded digital inputs: E44000 = in7..in0 connector X10. E44001 = in15..in8 connector X10. They can be read by part program and via dynamic operators. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E44100 to E44103	16 bits parameters to access the NCK embedded analogues inputs					
	I\O	Cmn	None	0..4095	Rpp	Rplc
	Rena	Rdop	Rdopc	UINT16	NObkp	NOrst
Description	Parameters used to access the NCK embedded analogue inputs: E44100 = Analogue in0 connector X9. E44101 = Analogue in1 connector X9. E44102 = Analogue in2 connector X9. E44103 = Analogue in3 connector X9. They can be read by part program and via dynamic operators. The relationship between value and output voltage is: 0V -> 0 +10V -> 4095 Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

8.5.5 Machining Parameters

E50000	Current tool correction number					
	Tools	Chnl	None	0..255	Rpp	NOplc
	Rena	Rdop	Rdopc	UINT8	NObkp	0
Description	Parameter related to function D. Example: current tool correction = Dxxx, i.e. E50000 = xxx.					

E51000	Current tool axis orientation					
	Tools	Chnl	None	0..102	RWpp	NOplc
	RWena	Rdop	Rdopc	UINT8	NObkp	NOrst
Description	Parameter giving the physical address of the axis parallel to the tool direction (G16...). If the tool axis orientation is negative, the value of 100 is added to the axis address, i.e.: 0 for X+ 100 for X- 1 for Y+ 101 for Y- 2 for Z+ 102 for Z- Example: E51000 = 2 (positive orientation on the Z axis), E51000 = 102 (negative orientation on the Z axis). Please also refer to the G16 on chapter 5.					

E50xxx	Tool length L of correction xxx or X dimension for Turning Machines					
	Tools	Cmn	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	This parameter corresponds to the values of the tool length or tool X dimension expressed in µm. xxx means from 001 to 255.					

E51xxx	Cutter tip radius @ of correction xxx or Z dimension for Turning Machines					
	Tools	Cmn	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	This parameter corresponds to the values of the tool corner radius or tool Z dimension expressed in µm. xxx means from 001 to 255.					

E52xxx	Tool radius R of correction xxx					
	Tools	Cmn	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	This parameter corresponds to the values of the tool radius expressed in µm. xxx means from 001 to 255.					

E53xxx	Dynamic correction DL or DX for tool offset xxx					
	Tools	Cmn	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	This parameter corresponds to the values of the dynamic tool correction in length (for milling) or diameter (for turning) . xxx means from 001 to 255.					

E54xxx	Dynamic radius correction DR (or DZ) for tool offset xxx					
	Tools	Cmn	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	This parameter corresponds to the values of the dynamic tool correction in radius (for milling) or in Z (for turning). xxx means from 001 to 255.					

E55xxx	Tool tip orientation for turning tools					
	Tools	Cmn	None	0..8	RWpp	NOplc
	RWena	RWdop	RWdopc	UINT8	NObkp	NOrst
Description	This parameter corresponds to the value of the tool tip orientation. xxx means from 001 to 255.					

E56xxx	Available parameters (H of the table of dynamic corrections)					
	Tools	Cmn	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	This parameter is used for management of tool wear or other information (maximum 8 characters). The values of H can be manipulated by the part program. For instance, if the program contains E56001 = E56001+1, the value of H of correction D1 is incremented by 1 each time the block is read. xxx means 001 to 255. The values of H can be entered manually (see Flexium HMI Operator Manual M00017)					

E57xxx	Parameter defining the tool type xxx					
	Tools	Cmn	P4N2	0..2	RWpp	NOplc
	RWena	RWdop	RWdopc	UINT8	NObkp	NOrst
Description	This parameter corresponds to the tool type. - 0: milling tool, - 1: turning tool, - 2: boring tool. xxx means from 001 to 255.					

E58xxx	Tools geometry code					
	Tools	Cmn	None	0...2^{32}-1	RWpp	NOplc
	RWena	RWdop	RWdopc	UINT32	NObkp	NOrst
Description	Unless it is used in the machine application, this is a free parameter which can be handled in a similar way as the E56xxx parameters (except that it can not be entered via HMI). This list of parameter is not used by the CNC software. It is available for the OEM. The general usage is for storing a tool geometry characteristic. xxx means 000 to 255.					

E6x000 E6x001 E6x004 E6x005	Geometric Transformations
Description	Parameters where x represents the axis number in the channel (0 to 8). The units of axes X, Y, Z, U, V and W (x = 0 to 5) and axes A, B and C (x = 6 to 8) are determined by the internal system units specified for the linear and rotary axes.

E6x000	DAT1 on axis x (x = axis address 0 to 8)					
	Axes	Chnl	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	These parameters corresponds to the DAT1 values in internal units (µm in general).					

E6x001	DAT2 on axis x (x = axis address 0 to 8)					
	Axes	Chnl	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	These parameters corresponds to the DAT2 values in internal units (µm in general)					

E6x002 and E6x003	Dynamic Machine Travel
Description	Parameters where x represents the axis number (0 to 8) in the channel. The units of axes X, Y, Z, U, V and W (x = 0 to 5) and axes A, B and C (x = 6 to 8) are determined by the internal system units specified for the linear and rotary axes.

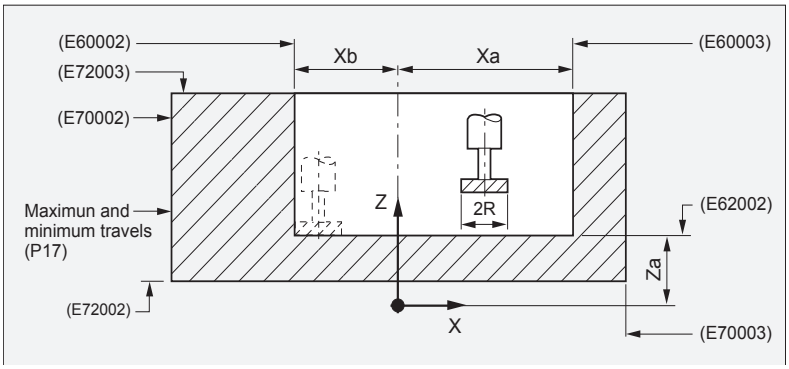
E6x002	Minimum dynamic machine travel on the axis "x"					
	Axes	Chnl	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst

Description

These parameters corresponds to the maximum dynamic machine axis travel in internal unit (µm in general) . The machine travels are constrained by software limits entered when setting up the machine (values contained in machine parameter P17 as indicated on Flexium Parameters manual M00021).

It may prove necessary to modify these limits according to the part to be machined or the environment (collision avoidance). Parameters E60002 and E60003 are used to modify the limits inside those defined by parameter P17.

At power on, after a reset or an M02, these parameters are initialised with the value 90000000 (+ or - according to whether it is the maximum or minimum).



The following programming is used to modify the limits:

```
%25
N..
E60002 = Xb     Declaration of the minimum dynamic travel on X
E60003 = Xa     Declaration of the maximum dynamic travel on X
E62002 = Za     Declaration of the minimum dynamic travel on Z
N..
N..
```

During machining, the system makes sure the tool tip remains within the limits defined. For instance, for maximum travel, the cutter axis cannot exceed dimension Xa-R.

E6x003	Maximum dynamic machine travel on the axis "x"					
	Axes	Chnl	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst

Description

These parameters corresponds to the maximum dynamic machine axis travel in internal unit (µm in general) . Please refer to E6x002 parameter for the description.

E6x004	Eccentricity offsets on axis "x" (DAT3)					
	Prg	Chnl	P4N2	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	These parameters are used to define the eccentricity values (equivalent to DAT3). The values are expressed in internal units (x=axis address from 0 to 5).					

E6x005	Offset programmed by G59 on axis "x"					
	Prg	Chnl	P4N2	+/- 99999999	RWpp	NOplc
	Rena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	These parameters are used to define the G59 offset along an axis. The values are expressed in internal units. Please also refer to the G59 on chapter 5.					

E69000	Scaling factor					
	Prg	Chnl	None	1...10000	RWpp	NOplc
	NOena	RWdop	RWdopc	UINT32	NObkp	NOrst
Description	These parameters are used to define the scaling factor. The values are expressed in thousands from 1/1000 up to 10000/1000.					

E69001	Inclined axis angle (grinders)					
	Prg	Chnl	1/10000 Deg	+/-800000	RWpp	NOplc
	NOena	RWdop	RWdopc	UINT32	NObkp	NOrst
Description	This parameter is used to define the angle beween X and Z axes on a grinder.					

E69002	Inclined wheel angle (grinders)					
	Prg	Chnl	1/10000	+/-800000	RWpp	NOplc
	NOena	RWdop	RWdopc	UINT32	NObkp	NOrst
Description	This parameter is used to define the wheel orientation angle on a grinder.					

E69003	Axis re-assignment					
	Axes	Chnl	None	0..888888	RWpp	NOplc
	NOena	RWdop	RWdopc	UINT32	NObkp	NOrst

Description

Axis re-assignment modifies the part programming three-axis reference system Xp Yp Zp with respect to the physical reference system of the machine, Xm, Ym, Zm to modify the programming reference system according to the workpiece position on the machine.

Axis re-assignment applies only to three-axis milling channels and can be forward (positive) or reverse (negative). It should be noted that axis assignment:

- Repositions the programmed dimensions, the interpolation plane (possibly reversing the direction of rotation of circles), the PQR material vectors, the coefficients of the polynomial functions, the 2D tool correction (possibly reversing G41/G42) and the tool direction (G16...)
- Is applied after programmed geometric transformations such as angular offset (ED..), programmed offset (G59..), mirroring (G51..) and scaling factor (G74).

Axis assignment does not apply to dimensions programmed with respect to the measurement origin (G52..). Datum shifts DAT1, DAT2 and DAT3 are not affected by axis assignment. When programming G16.., the programmed tool direction is repositioned according to the axis assignment and, after a system reset, assumes the assignment of the program Z.

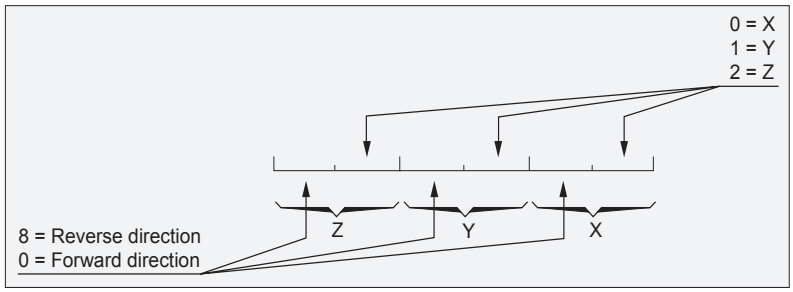
Axis assignment by parameter E69003

Parameter E69003 is used to read the current assignments or declare new ones. This parameter includes three words:

- The units and tens digits contain the X axis assignment,
- The hundreds and thousands digits contain the Y axis assignment,
- The ten thousands and hundred thousands digits contain the Z axis assignment.

In each word, the value 0 addresses the X machine axis, the value 1 addresses the Y machine axis and the value 2 addresses the Z machine axis to which is added the value 80 if the assignment is reversed.

Example:
Axis assignment XR+ YQ- ZP- is declared by E69003 = 808102.



Remarks

The tool direction (G16..) in the program reference system must be declared again after changing the axis assignments by programming parameter E69003.

A format error in programming of E69003 returns error message 94.

Programming of turning planes G20, G21 and G22 with reassigned axes returns error message 2.

Programming of E69003 with a turning plane not disabled returns error message 95.

Please also refer to the Gxx on chapter 5.

E69004	Tool disengagement in G41/G42 for an angle bigger that 120°					
	Prg	Chnl	P4N2	0..999...	RWpp	NOplc
	NOena	RWdop	RWdopc	UINT32	NObkp	NOrst

Description

This parameter is used to prevent contact between the tool and the part at the angle tip.

Please also refer to the G41/G42 on chapter 5.

8.5.6 Channel Axis Access Parameters

E7x000 to E7x007 - E7x100, E7x101	Parameters where "x" represents the axis name in the channel (0 -> X, 1 -> Y .. 8 -> C)
Description	For parameters E7x000 to E7x004, the units of axes X, Y, Z, U, V and W (x = 0 to 5) and axes A, B and C (x = 6 to 8) are determined by the internal system units defined for the linear and rotary axes. (See also chapter 4.1).

E7x000	Position reference of an axis in the channel					
	Axes	Chnl	P4N2	+/- 99999999	Rpp	NOplc
	Rena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	This parameters defines the axes reference as generated by the interpolator. The values are expressed in internal units. Read-only parameters (except for dynamic operators) Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E7x001	Sampling the axes positions for on-the-fly probing					
	Axes	Chnl	P4N2	+/- 99999999	RWpp	NOplc
	Rena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	These parameters are used to directly read the axes position during movement after an NC priority interrupt (Probing input) programmed by function G10. The measurement is stored in the memory until: - the next interrupt is received - a manual or programmed (M02) reset is carried out. When function G10 is read in the part program, these parameters should be preset to a particular value by programming to make a test when acquiring the dimension (if there is no interrupt before the end of block, then this particular value can be detected). Please also refer to the G10 on chapter 5.					

E7x002	Minimum static machine travel					
	Axes	Chnl	P4N2	+/- 99999999	Rpp	NOplc
	Rena	Rdop	Rdopc	SINT32	NObkp	P17 / NOrst
Description	This parameter is the image of machine parameter P17 (see also E6x002 and E6x003). Please refer to machine parameter P17 on Flexium Parameters manual M00021.					

E7x003	Maximum static machine travel					
	Axes	Chnl	P4N2	+/- 99999999	Rpp	NOplc
	Rena	Rdop	Rdopc	SINT32	NObkp	NOrst
Description	This parameter is the image of machine parameter P17 (see also E6x002 and E6x003). Please refer to machine parameter P17 on Flexium Parameters manual M00021.					

E7x004	Direction of movement of the axes being interpolated					
	Axes	Chnl	P4N2	+/- 99999999	Rpp	NOplc
	Rena	Rdop	Rdopc	SINT32	NObkp	NOrst
Description	This parameter is used only in conjunction with dynamic operators. With linear interpolation, this parameter contains the incremental move for the current block on the relevant axis. With circular interpolation, this parameter contains the component on axis x of the vector with length R tangent to the path (applies only to the linear axis of the interpolation plane).					

E7x005	Axis address assignment					
	Axes	Chnl	None	-1/0...31	RWpp	NOplc
	Rena	Rdop	Rdopc	SINT8	NObkp	P9

Description

Read of these parameters gives the physical address of a program axis (a program axis is assigned to a physical axis by machine parameter P9).

When no physical axis is assigned to the program axis, the parameter takes on the value -1.

Examples:
- if word N2 of parameter P9 has the value 05 (program axis 5=W of channel 1 assigned to physical address 2) in the first channel, E75005=2 (physical axis 2 assigned to program axis 5),
- if word N8 of parameter P9 has the value 26 (program axis 6=A of channel 3 assigned to physical address 8) in the third channel, E76005=8 (physical axis 8 assigned to program axis 6),
- if no word of P9 has the value 13 (program axis 3=U of channel 2 is unassigned) in the second channel, E73005=-1 (no physical axis assigned to program axis 3).

Writing to these parameters can assign a different physical axis to a program axis.

A physical axis must be freed before being able to be assigned to a new program axis (by assigning the value -1 to the parameter of the program axis with which the physical axis was originally associated).

Example:
- in a channel, E75005=2 (physical axis 2 assigned to program axis 5=W),
- in the same channel, E74005=8 (physical axis 8 assigned to program axis 4=V),
- to assign physical axis 8 to program axis 5 (W):
  - release physical axis 8 from program axis 4 (E74005=-1),
  - assign physical axis 8 to program axis 5 (E75005=8).

A new physical axis assigned to a program axis must be of the same nature as the previous axis (linear or rotary axis, servo or not, modulo or with limited excursion, etc.).

A reset or end of program (M02) restores the default conditions for the axes setting.

Please refer to machine parameter P9 on Flexium Parameters manual M00021.

E7x006	Axis coupling					
	Axes	Chnl	None	0/1	RWpp	NOplc
	Rena	Rdop	Rdopc	UINT8	NObkp	P64
Description	Parameters used to create or cancel carrier/carried axis pairs. These parameters apply to primary or secondary linear axes (program axis number between 0 and 5). The value «0» assigned to the parameter cancels the primary/secondary axis pairing to which the axis specified belongs and a «1» specifies that the axis is carried. When the parameter related to an axis is modified, the parameter related to the other axis in the primary/secondary axis pair is automatically modified. A reset reconfigures the axis pairs as specified by machine parameter P64. Example: E70006 = 0 Axes X and U are independent E72006 = 1 Axes Z and W are coupled. Please refer to machine parameter P64 on Flexium Parameters manual M00021.					

E7x007	Axes programmed by diameter					
	Axes	Chnl	None	0/1	RWpp	NOplc
	Rena	Rdop	Rdopc	UINT8	NObkp	P4
Description	Only parameters E70007 (X axis) and E73007 (U axis) are programmable by diameter. These parameters are reset for programming by radius and set for programming by diameter.					

E7x100	Interpolator position references					
	Axes	Chnl	P4N2	+/- 99999999	Rpp	NOplc
	Rena	Rdop	Rdopc	SINT32	NObkp	NOrst
Description	This parameter contains the position references from the interpolators. Interaxis calibration uses these references after transformation for computing corrections. The interpolated references E7x100 (with or without transformation) are copied into the references used in servo-controls (E7x000) only for the axes declared in machine parameters P2 and P3 (possibly modified by E9100x). Please refer to machine parameter P2 and P3 on Flexium Parameters manual M00021.					

E7x101	Interpolation speed limiting					
	Axes	Chnl	None	0...100	RWpp	NOplc
	Rena	NOdop	RWdopc	UINT8	NObkp	100

Description

Write of this parameter limits the interpolation speed on logical axis x in the channel where it is programmed according to the relation:

$$V\ ipo\ (x) = P30 * \frac{E7x101}{100}$$

However, the maximum speed on the axis is not modified.
The limit value accepted is between 0 and 100 (percent) (rounded off to the nearest integer).
These parameters are forced to 100 percent at power on. They are all reset to 100 percent after a reset.

Possible uses of this parameter:

- Typical use: reserves part of the speed for geometric transformations carried out downstream of the interpolation, for instance with RTCP with or without axis driven in NM_AUTO mode (see Flexium Extended programming manual M00020EN).

- Special use with E7x101 = 0: axis x is assigned a zero speed:
  - if it is programmed alone in a block, it is ignored
  - if it is programmed with other axes, only the movements on the other axes are carried out.

It should be noted that:
- programming a non-CNC channel or programming a non-existent axis returns error message 91
- if the programmed value is negative or above 100, the system returns error message 92.

E79002	Value of the feed rate potentiometer assigned to the channel					
	Axes	Chnl	None	0...255	Rpp	NOplc
	Rena	RWdop	RWdopc	UINT8	NObkp	NOrst

Description

These parameters are used to address the feed rate override coefficient.
(E79002 = 128 corresponds to the 100 percent setting of the potentiometer).

This parameter can only be modified by dynamic operators (read-write).

E79003	Distance remaining to be travelled in the block being executed					
	Axes	Chnl	P4N2	+/- 99999999	Rpp	NOplc
	Rena	RWdop	RWdopc	SINT32	NObkp	0

Description

The distance on the path programmed in the block being executed is expressed in the internal unit of the system.

This parameter can only be modified by dynamic operators (read-write).

Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".

The CNC can in some cases break down the block in smaller ones (e.g. RTCP). In such a case E79003 refers to the distance remaining in the smaller blocks.

E79004	Current feed rate on the path programmed in the block					
	Axes	Chnl	None	0..32767	Rpp	NOplc
	Rena	RWdop	RWdopc	UINT32	NObkp	0
Description	The value is expressed in internal system unit by sampling. This read-only parameter can only be modified by dynamic operators. Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E79005	Minimal value for feed override					
	Axes	Chnl	None	0.. 255	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT8	NObkp	P67
Description	This parameter is used to define the minimal value of the feed rate override.					

E79006	Maximum value for feed override					
	Axes	Chnl	None	0.. 255	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT8	NObkp	P67
Description	This parameter is used to define the maximum value of the feed rate override.					

E79007	Minimal value for spindle override					
	Spindle	Chnl	None	0.. 255	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT8	NObkp	P67
Description	This parameter is used to define the minimal value of the spindle speed override.					

E79008	Maximum value for spindle override					
	Spindle	Chnl	None	0.. 255	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT8	NObkp	P67
Description	This parameter is used to define the maximum value of the spindle speed override.					

E79009	Distance to go within the block					
	Axes	Chnl	None	+/- 99999999	Rpp	NOplc
	Rena	Rdop	Rdopc	SINT32	NObkp	0
Description	This parameter is used to find the distance to be travelled within the block. It is generally used in conjunction with E79010 to determine the percentage of execution of a block.					

E79010	Distance remaining in the current programmed block					
	Axes	Chnl	P4N2	+/-99999999	Rpp	NoPlc
	NOEna	Rdop	Rdopc	SINT32	Nobkp	0
Description	This parameters gives the distance remaining in the current programmed block. It is used in conjunction with E79009 to compurte the percentage of execution of a block. This parameter is very similar to E79003 except that E79003 indicates the distance per broken down blocks when they are some (e.g. RTCP).					

E79011	Expansion factor					
	All	Chnl	None	+/- 999999…	Rpp	NoPlc
	NOEna	Rdop	Rdopc	SINT32	Nobkp	0
Description	The CNC may in some case 'zoom' the interpolation data to increase precision. This parameters gives the expansion factor.					

8.5.7 Spindle Access Parameters

E79000	Reference position of the currently measured spindle					
	Spindle	Chnl	None	0...3599999	Rpp	NOplc
	Rena	RWdop	RWdopc	UINT32	NObkp	NOrst
Description	Position in ten-thousandths of a degree. When the position is accessed by dynamic operators, it is expressed as 1/4096 revolution modulo 2^{16}. Read-only parameters (except via the dynamic operators).					

E79001	Speed demand for the currently controlled spindle					
	Spindle	Chnl	None	+/-32767	Rpp	NOplc
	Rena	RWdop	RWdopc	SINT16	NObkp	0
Description	The value is: (speed demand/max speed of the range) x 2^{15}. In «normal» parametric programming, this parameter is a read-only parameter and gives the value programmed by «S» as a function of the speed range, but not the value calculated by the dynamic operators. The reference value is encoded on 15 bits + sign bit. Read-only parameters (except via the dynamic operators). When this parameter is programmed by dynamic operators, the sign bit is used by the system only in state M05. Otherwise, it is forced to insure the direction of rotation programmed by M03 or M04. The spindle potentiometer is not active in this case.					

E9010x	Spindle x position reference					
	Spindle	Cmn	None	0...3599999	Rpp	NOplc
	Rena	RWdop	RWdopc	UINT32	NObkp	NOrst
Description	This parameter allows all the channels to access all the spindles. Same format as parameter E79000. Read-only parameters (except by dynamic operators) where «x» is the spindle number (0 to 3).					

E9011x	Spindle x modulo					
	Spindle	Cmn	1/10000 Deg	1...3600000	Rpp	NOplc
	Rena	RWdop	RWdopc	UINT32	NObkp	NOrst
Description	This parameter allows all the channels to access all the spindles. Same format as parameter E79000. Read-only parameters (except by dynamic operators) where «x» is the spindle number (0 to 3). For additional information please refer to machine parameter P40 on Flexium Parameters manual M00021 and also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

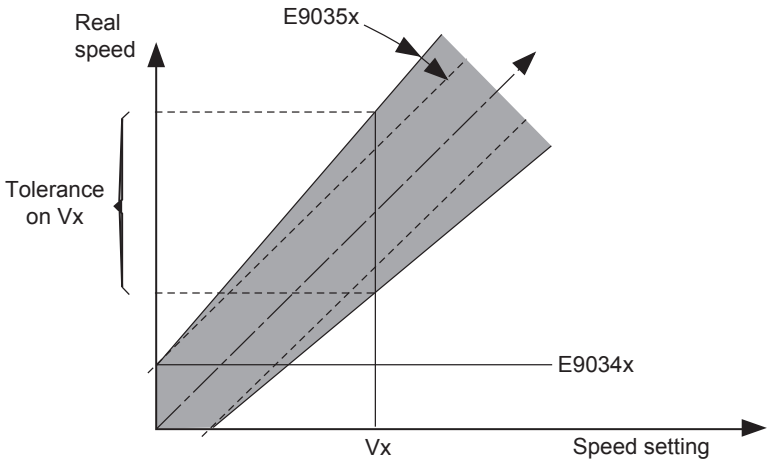
E9020x	Spindle x speed setting					
	Spindle	Cmn	None	+/- 32767	Rpp	NOplc
	Rena	RWdop	RWdopc	UINT32	NObkp	0
Description	This parameter allows all the channels to access all the spindles. Same format as parameter E79001. Read-only parameters (except by dynamic operators) where «x» is the spindle number (0 to 3). Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators".					

E9030x	Maximum spindle x indexing speed					
	Spindle	Cmn	Rpm	0...65535	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT16	NObkp	P43
Description	Read/write parameters where «x» is the spindle number (0 to 3). The new values set for these parameters are saved until the next NC reset. Value expressed Speed in rev/min. Please refer to machine parameter P43 on Flexium Parameters manual M00021.					

E9031x	Spindle x in-position window					
	Spindle	Cmn	1/10000 Deg	0...65535	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT16	NObkp	P44N0..3
Description	Read/write parameters where «x» is the spindle number (0 to 3). The new values set for these parameters are saved until the next NC reset. Value expressed in internal units. Please refer to machine parameter P40 and P44 on Flexium Parameters manual M00021.					

E9032x	Spindle x indexing gain					
	Spindle	Cmn	Rpm/rev	0...65535	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT16	NObkp	P45N0..3
Description	Read/write parameters where «x» is the spindle number (0 to 3). The new values set for these parameters are saved until the next NC reset. Value expressed in rev/min/rev. Please refer to machine parameter P45 on Flexium Parameters manual M00021.					

E9033x	Spindle x indexing acceleration					
	Spindle	Cmn	P6N1	0...65535	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT16	NObkp	P32N48...N55
Description	Read/write parameters where «x» is the spindle number (0 to 3). The new values set for these parameters are saved until the next NC reset. Value expressed in Deg/s^2. Please refer to machine parameter P32 on Flexium Parameters manual M00021.					

E9034x to E9035x	Spindle speed control parameters
Description	These parameters are used to make sure the difference between the spindle speed set and the real speed remains within a tolerance interval and also, where required, to detect nonrotation of the spindle when the speed is below a minimum threshold. 

E9034x	Speed threshold below which spindle x is considered stopped					
	Spindle	Cmn	Rpm	0.. 65000	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT16	NObkp	NOrst
Description	Parameter defining the speed threshold below which the spindle is considered to be stopped. Read/write parameter where «x» is the spindle number (0 to 3). This parameter is expressed in rpm. It is initialised with 10 at power on and can be modified by programming. This parameter is also used as tolerance threshold and detection of the speed reached.					

E9035x	Tolerance interval coefficient					
	Spindle	Cmn	1/256	0...255	RWpp	NOplc
	Rena	RWdop	RWdopc	UINT16	NObkp	NOrst
Description	Parameter defining the tolerance interval coefficient as a function of the speed. Read/write parameter where «x» is the spindle number (0 to 3) This parameter expressed as 1/256 is initialised with 13 (i.e. 5 percent) at power on and can be modified by programming.					

8.5.8 Common NC Parameters

E8x000 to E8x999	Parameters related to the inter-axis calibration
Description	These parameters (32 bits) are related to the inter-axis calibration. (Please also refer to Flexium Installation manual M00009) If they are not used for inter-axes calibartion they are free to use for any other purpose. Whereas E80xxx, E81xxx and E82xxx parameters are backed up the E83xxx and E84xxx parameters are not saved during power down of the system.

E80000 to E80199	Local 32 bits parameters					
	All	Cmn	None	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	YESbkp	NOrst
Description	Generic purpose parameters. Backed-up.					

E81000 to E81999	Master axis position reference					
	All	Cmn	None	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	YESbkp	NOrst
Description	Interaxes calibration tables - master axes data. These parameters can also be used as generic purpose parameters.					

E82000 to E82999	Slave axis corrections with respect to the master axes					
	All	Cmn	None	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	YESbkp	NOrst
Description	Interaxes calibration tables - slave axes data. These parameters can also be used as generic purpose parameters.					

E83000 to E83999	General purpose 32 bits parameters					
	All	Cmn	None	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	Generic purpose parameters. These parameters are not backed-up.					

E84000 to E84999	**General purpose 32 bits parameters**					
	All	Cmn	None	+/- 99999999	RWpp	NOplc
	RWena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	Generic purpose parameters. These parameters are not backed-up.					

8.5.9 Machine Axis Access Parameters

These parameters are generally set by the machine builder according to the machine characteristics. Do not attempt to modify them if you are not familiar with the setting and tuning of the axes. (xx = 0 to 31)

E9yyxx	Parameters for physical address of the axis (0 to 31)
Description	Parameters where xx represent the physical address of the axis (0 to 31)

E900xx	Axis measurement					
	Axes	Cmn	P4N2	+/- 99999999	RWpp	NOplc
	Rena	RWdop	RWdopc	SINT32	NObkp	NOrst
Description	Read on measured or servo axes. Write is possible only for non-servo axes.					

E910xx	Servo or non-servo status of the axes					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	P3..N0
Description	These parameters reflect the servo or non-servo status of an axis (a physical axis is declared servo or non-servo by machine parameter P3) and are used to modify this state. A «1» in this parameter indicates that the corresponding axis is servo controlled and a «0» that it is non-servo. A reset or end of program (M02) reconfigures the axes as specified by machine parameter P3. Please refer to Flexium Parameters manual M00021.					

E911xx	Homing status on an axis					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	NOrst
Description	These parameters indicate whether the axis homing procedure has been done. A «1» indicates that a HOME has not been completed on the axis and a «0» that it has been completed. This parameter addresses the axes and spindles, e.g.: If spindle 1 is measured, programming E91124 = 1 forces no homing for the spindle. Since homing is automatic on the spindles, this makes it possible to reset the spindle position reference.					

E912xx	N/M AUTO axes					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	0
Description	These parameters define the axes concerned by the N/M AUTO function. Use of the list of E912xx parameters concerns only the servo-controlled axes (error message 99 is generated by an attempt to assign an axis that is not servo-controlled). Parameters E912xx cannot use the PLC axes. The number of parameters that can be used is limited to five (error message 99 is generated if this number is exceeded). When the N/M AUTO function is enabled, the axes concerned by this function are no longer controlled by the interpolators. However, the interpolator computation is continued in parallel to allow the path to be resumed after inhibiting of function N/M AUTO. Only a parameter E912xx belonging to the channel can be written (error message 99 is generated by an attempt to write E912xx for a channel different from the one of the part program in which it is programmed). After a reset, parameters E912xx are reset to zero. When a parameter E912xx is reset, the part program in which it is programmed is stopped if the corresponding N/M AUTO axis is being moved by an axis jog or handwheel. Execution of the program is not resumed until the axis is stopped. For additional information on the N/M AUTO function, refer to the Flexium Extended Programming manual M00020.					

E913xx	Clampable axis enable state					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	P8..N0
Description	This parameter enables or disables axis clamping by programming. Example: - E91308 = 1: axis 08 can be clamped - E91308 = 0: axis 08 cannot be clamped. A reset applies the settings of machine parameter P8 to the axes. The list of axes that can be clamped is sent to the PLC (by PLC variable RCNC.Axis[i].Clamp).					

E920xx	Axis origin switch enable state					
	Axes	Cmn	None	0/1	Rpp (Wpp P3)	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	0
Description	A «1» indicates that the switch is enabled and a «0» that it is inhibited. The state of the control word can be read on all the measured axes of the machine. Writing to this control word is possible only for non-servo axes. (see P3 parameters M00021) A reset, end of program (M02) or contact with the measurement sensor zero automatically cancels inhibiting of the switches. In HOME mode, it takes on a value of «1» when the axis jog is engaged and «0» when it is released or when the switch is encountered.					

E922xx	Scaling the lag display					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	NOrst
Description	When set to 1 the display of the lag is divided by 10 to harmonize the display if axes with very different characteristics are used.					

E930xx	Axis @xx home switch status					
	Axes	Cmn	None	0/1	Rpp	NOplc
	NOena	NOdop	NOdopc	....	NObkp	NOrst
Description	Status of the home switch for the axes. A "1" indicates that the corresponding switch is open.					

E931xx	Measured axis					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	P2.. N0
Description	A «1» indicates that the axis is measured. This parameter is the image of machine parameter P2. Please refer to Flexium Parameters manual M00021.					

E932xx	Axis declared as rotary modulo 360 degrees					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	P1.. N0
Description	A «1» indicates that the axis is declared rotary modulo 360 degrees. This parameter is the image of machine parameter P1 Word N1. Please refer to Flexium Parameters manual M00021.					

E933xx	Axis homing direction					
	Axes	Cmn	None	0/1	Rpp	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	P15N10
Description	A «1» indicates that the axis is moving in the negative direction in homing mode. This parameter is the image of machine parameter P15 Word N0. Please refer to Flexium Parameters manual M00021.					

E934xx	Homing status without switch wiring					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	RWdopc	UINT32	NObkp	P15N1
Description	A «1» indicates that the origin switch is not wired on this axis. This parameter is the image of machine parameter P15 Word N1. Please refer to Flexium Parameters manual M00021.					

E935xx	Axis (or spindle) in-position window					
	Axes	Cmn	None	0/1	Rpp	Rplc
	NOena	NOdop	RWdopc	UINT32	NObkp	NOrst
Description	This parameter is used to define the position of an axis or a spindle. Its state is the image of the bits of RCNC.Axis[i].Moving transmitted to the PLC. However, if physical axis numbers 24 to 27 correspond to spindles, the associated parameters, E93524 to E93527, indicate the spindle in-position status (spindle indexed by M19 or synchronised spindle) and are the image of the bits of RCNC.Spindle[s].PositionReached transmitted to the PLC.					

E940xx	Parameters related to inter-axis calibration					
	Axes	Cmn	None	-1/0...31	RWpp	NOplc
	NOena	NOdop	RWdopc	SINT32	NObkp	NOrst
Description	They are used to define the relationship between slave and master. For additional information about inter axes calibration please refer to the Flexium Commissioning manual M00010.					

E941xx	Association of a slave axis (or spindle) with a master axis (or spindle)					
	Axes	Cmn	None	-1/0...31	RWpp	NOplc
	NOena	NOdop	RWdopc	SINT32	NObkp	P27
Description	The axes must be associated before slaving an axis to a master axis for duplication or synchronisation. Please refer to the Flexium Commissioning manual M00010 "Duplicated and synchronised axes". The associations can be defined by machine parameter P27 or in the part program, e.g.: E941xx = m The slave axis is denoted «xx» and the master axis «m» If «m» = -1, the association of slave axis «xx» with master axis «m» is cancelled. For the slave and master axis are displayed in "Duplicated axes" page of HMI. please refer to Flexium HMI Operator manual M00017 For the machine parameter P27 please refer to Flexium Parameters manual M00021.					

E950xx	Axis reference correction					
	Axes	Cmn	P4 N2.. N4	+/- 99999999	Rpp	NOplc
	NOena	RWdop	RWdopc	SINT32	NObkp	0
Description	These parameters can only be written by interaxis calibration or dynamic operators. Please refer to Flexium Commissioning Manual M00010, or by the dynamic operators as indicated on the Flexium Extended Programming manual M00020.					

E951xx	Position of the origin switch with respect to the machine origin					
	Axes	Cmn	None	+/- 99999999	Rpp	NOplc
	NOena	NOdop	RWdopc	SINT32	NObkp	P16.. N0..
Description	This parameter is the image of machine parameter P16. Please refer to Flexium Parameters manual M00021.					

E952xx	Axis (or spindle) measurement correction					
	Axes	Cmn	P4 N2.. N4	+/- 99999999	Rpp	NOplc
	NOena	Rdop	RWdopc	SINT32	NObkp	0
Description	This parameters returns the current correction of the measure. This correction is the sum of corrections due to backlash and axis calibration.					

E953xx	Differential offset value					
	Axes	Cmn	P4 N2.. N4	+/- 99999999	Rpp	NOplc
	NOena	Rdop	RWdopc	SINT32	NObkp	0
Description	Read only parameter returning the amount of differential offset generated by Dynamic operators code 24. Please also refer to Flexium Extended Programming manual M00020.					

E960xx	Duplicated axis in automatic modes					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	These parameters are used in case of axes duplication and synchronisation. When set to a '1' the axis is duplicated in the automatic modes. If it is a «0», it is not.					

E961xx	Duplicated axis in manual mode (JOG)					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	These parameters are used in case of axes duplication and synchronisation. When set to a '1' , the axis is duplicated in manual mode (JOG). If it is a «0», it is not.					

E962xx	Synchronised axis					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	These parameters are used in case of axes duplication and synchronisation. When set to a '1', the axis is synchronised. If it is a «0», it is not.					

E963xx	Symmetrically driven axis					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	These parameters are used in case of axes duplication and synchronisation. When set to a '1', the axis is driven symmetrically. If it is a «0», it is driven in the same direction as the master.					

E970xx	Maximum axis speed					
	Axes	Cmn	mm/min or deg/min	99999..	Rpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter defines the maximum speed.					

E971xx	Axis acceleration at programmed feed rate					
	Axes	Cmn	mm/s^2 or deg/s^2	99999..	Rpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter defines the acceleration expressed in mm/s^2 or deg/s^2.					

E972xx	Axis acceleration in Rapid					
	Axes	Cmn	mm/s^2 or deg/s^2	99999..	Rpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter defines the acceleration expressed in mm/s^2 or deg/s^2.					

8.5.10 Axis adjustment

E923xx	Enable or disable Phase lag and Phase lead compensation					
	Axes	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parametert is used to validate or inhibit the phase lag and lead filter. For additional information please refer to the Flexium Extended Programming manual M00020.					

E973xx	Maximum authorised speed at angle crossings					
	Axes	Cmn	P4 N2.. N4	99999..	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter is the image of machine parameter P33. Its value is specified in mm/minute. Please refer to Flexium Parameters manual M00021.					

E981xx	Axis acceleration anticipation time constant					
	Axes	Cmn	µs	99999..	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	When machining at very high speed, this parameter is used to program the axis acceleration anticipation. After an initialisation, the value of machine parameter P19 is reused. Please refer to Flexium Parameters manual M00021.					

E982xx	Amplitude of the antistick pulse on reversal					
	Axes	Cmn	P4 N2.. N4	99999..	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter is the image of machine parameter P19 (words N32 to N63). It is expressed in micrometres. Please refer to Flexium Parameters manual M00021.					

E983xx	Time constant to absorb the antistick pulse					
	Axes	Cmn	1/100ms	99999..	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter is the image of machine parameter P19 (words N64 to N95). Its value is expressed in 1/100000 s. Please refer to Flexium Parameters manual M00021.					

E984xx	Delay to apply the drive reference					
	Axes	Cmn	None	99999..	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter is not used for machining. It is intended for calculation purposes. It is used by NUM for some internal macros.					

E98500 to E98563	Partial feed forward per axis					
	Axes	Cmn	1/256	0..512	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter can be used to adjust the lag values with axes having very different characteristics. The values are expressed in 1/256. The first set of 32 values defines the partial feed forward outside HSC the second set with HSC validated.					

E98600 to E98663	Anticipation time constant (first 32 values) and Lag balancing error filter time constant (last 32 values)					
	Axes	Cmn	1/100ms	0..512	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter is used to adjust anticipation and averaging filter in HSC.					

8.5.11 Special purpose parameters

8.5.11.1 Delayed power down on machine errors

These parameters are used to delay the power down in case of excess lag error, for example on a grinding or gear hobbing machine we may want to retract the wheel before shutting down the axes.

These parameters are managed by the machine builder application. Do not attempt to modify them if you are not familiar with the machining cycle and safeties.

E92100 to E92131	Axis error masking					
	Axes	Cmn	None	0/1	Rpp	NOplc
	NOena	RWdop	RWdopc	UINT32	NObkp	0
Description	Delayed power down masks the error display and power fail if the corresponding bit is set. These parameters are intended to be used with dynamic operators in order to mask a machine error on an axis long enough to take any appropriate action (tool disengagment for example).					

E92400 to E92403	Spindle error masking					
	Spindle	Cmn	None	0/1	Rpp	NOplc
	NOena	RWdop	RWdopc	UINT8	NObkp	0
Description	Delayed power down masks the error display and power fail if the corresponding bit is set. These parameters are intended to be used with dynamic operators in order to mask a machine error on an spindle long enough to take any appropriate action (tool disengagment for example).					

E93700 to E93731	Indication of axis in error or not					
	Axes	Cmn	None	0/1	Rpp	NOplc
	NOena	Rdop	Rdopc	UINT32	NObkp	0
Description	These parameters define which axes (if any) have detected a machine error (excess lag).					

E93800 to E93803	Indication of spindle in error or not					
	Spindle	Cmn	None	0/1	Rpp	NOplc
	NOena	Rdop	Rdopc	UINT8	NObkp	0
Description	These parameters define which spindles (if any) have detected a machine error (excess lag).					

8.5.11.2 Axis / Spindle switching parameters

The following sets of parameters are used.

E90120 to E90123	C axis / Spindle measurement conversion used for Spindle/ axis switching					
	Axes	Cmn	None	+/-99999..	Rpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This is a read only parameter used to display the coefficient between the axis measure increment and the spindle measure increment in case of Spindle /Axis switching. It is intended for verification. For additional information please refer to Spindle/Axis switching in the Flexium Commissioning manual M00010.					

E942xx	Association between drive and axis or spindle command					
	Axes	Cmn	None	0..31	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	P70N8
Description	This parameter is used to switch either the Spindle or axis command to the same drive in case of spindle/axis commutation. This parameter defines if the spindle xx (24>27) is controlled as a spindle (value 24> 27) or the associated axis. This parameter is mostly used for Spindle / Axis switching. For additional information about this feature please refer to the Flexium Commissioning manual M00010.					

E943xx	Physical handwheel coupler in the list of handwheels					
	Axes	Cmn	None	0 ..3	RWpp	NoPlc
	Rena	Nodop	Rdopc	UINT32	Nobkp	P14 / P70
Description	This parameter is used to switch an axis analog output to an handwheel coupler.					

E353xx	Select a set of drive parameters					
	Axes	Cmn	None	00/01/10/11	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	This parameter is used to select on the drive a set of Speed or Position loop parameters among the two possibles for each case.					

Additional information about Axis / Spindle switching will be found in the Flexium Commissioning manual M00010.

8.5.11.3 Spindle synchronisation

The following parameters are used for spindle synchronisation:

E90360 to E90363	Enable synchronisation on master spindle					
	Spindle	Cmn	None	0.. 65000	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	These parameters are used in spindle synchronisation. For additional information about the use of these parameters will be found in the Flexium commissioning manual M00010 "Spindle synchronisation". For the master spindle, this parameter when >0 validates a time constant modification according to the speed. For the slave spindles this parameter defines (in 1/1000) the target of time constant reduction at high speeds. e.g. 1200 T1 = T0 /1.2 (T1 time constant at high speeds, T0 at low speeds)					

E90370 to E90373	Spindle rotation speed for maximum torque					
	Spindle	Cmn	Rpm	0/65000	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	Information about the use of these parameters will be found in the Flexium commissioning manual M00010 "Spindle synchronisation".					

E90380 to E90383	Spindle rotation speed for maximum power					
	Spindle	Cmn	Rpm	0/65000	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	Information about the use of these parameters will be found in the Flexium commissioning manual M00010 "Spindle synchronisation".					

E94124 to E94127	Master spindle number for spindle synchronisation					
	Spindle	Cmn	Rpm	-1 or 24..27	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	Information about the use of these parameters will be found in the Flexium commissioning manual M00010 "Spindle synchronisation".					

E98024 to E98027	Spindle integral action time constant					
	Spindle	Cmn	ms	0/65000	RWpp	NOplc
	NOena	NOdop	NOdopc	---	NObkp	NOrst
Description	0: No integral action >0: Integral action <0: Integral action frozen. Information about the use of these parameters will be found in the Flexium commissioning manual M00010 "Spindle synchronisation".					

E90224 to E90227	Spindle synchronisation errors					
	Spindles	Cmn	P4..N4	0 ..3	Rpp	NoPlc
	Rena	Nodop	Nodopc	UINT8	Nobkp	0
Description	This parameters shows the synchronisation error for the spindles 1 (E90224) to 4 (E90227).					

8.5.11.4 Drive access parameters

The following parameters are used to access drive test points via dynamic operators.

Please also refer to Flexium Extended Programming manual M00020 "Dynamic operators"

E35100 to E35131	First 16 bits test point value					
	Drive	Cmn	None	0...2^{16}-1	Rpp	NOplc
	NOena	Rdop	Rdopc	UINT16	NObkp	NOrst
Description	These parameters define the first 16 bits test point value for drive 00 (E35100) to 31 (E35131) selected by V130 in this drive. Please also refer to NUMDrive C Parameters manual AMOMAN007 "V130 Associate MP with TP1".					

E35132 to E35163	Second 16 bits test point value					
	Drive	Cmn	None	0...2^{16}-1	RWpp	NOplc
	NOena	Rdop	Rdopc	UINT16	NObkp	NOrst
Description	These parameters define the second 16 bits test point value for drive 00 (E35132) to 31 (E35164) selected by V131. Please also refer to NUMDrive C Parameters manual AMOMAN007 "V131 Associate MP with TP2".					

E35164 to E35195	32 bits test point value					
	Drive	Cmn	None	0...2^{32}-1	Rpp	NOplc
	NOena	Rdop	Rdopc	UINT32	NObkp	NOrst
Description	These parameters define the 32 bits test point value for drive 00 (E35164) to 31 (E35195). Overwrites any 16 bits value.					

E35200 to E35231	Code number of the first test point					
	Drive	Cmn	None	0..500	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	NOrst
Description	These parameters define the code number of the first test point to read in drive 00 to 31 equivalent to V130 drive parameter. Please also refer to NUMDrive C Parameters manual AMOMAN007 "V130 Associate MP with TP1".					

E35232 to E35263	Code number of the second test point					
	Drive	Cmn	None	0..500	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	NOrst
Description	These parameters define code number of the second tests point to read in drive 00 (E35232) to 31 (E35263) equivalent to V131 drive parameter. Please also refer to NUMDrive C Parameters manual AMOMAN007.					

E35264 to E35295	Enable the reading test point					
	Drive	Cmn	None	0/1	RWpp	NOplc
	NOena	NOdop	NOdopc	...	NObkp	NOrst
Description	These parameters when set to 1 enable the reading the test point in drive 00 (E35264) to 31 (E35295). Please also refer to NUMDrive C Parameters manual AMOMAN007.					

Page deliberately empty

Page deliberately empty

Summary

9 List of Errors 539
9.1 Miscellaneous Errors and Machine Errors 539
9.2 Parametric Programming Errors 542
9.3 Profile Geometry Programming (PGP) Errors 543
9.3.1 The end point cannot be calculated from the elements in the blocks 543
9.3.2 An intermediate point cannot be calculated from the data in two blocks 543
9.3.3 An intermediate point cannot be calculated from the data in three blocks 543
9.3.4 Fillet or chamfer definition errors 543
9.3.5 Miscellaneous errors in PGP 543
9.4 Miscellaneous Errors 544
9.5 Request for Movements Outside the Machine Travel Limits 544
9.6 Structured Programming Errors 544
9.7 Axis Errors 545
9.8 CNC - PLC Communication 545
9.9 System options errors 545
9.10 Axes Not Identified on the Bus 545
9.11 Dynamic Operators in C 546
9.12 Spline Curve Interpolation Errors 546
9.13 Cycle Programming Errors 547

Page deliberately empty

9 List of Errors

9.1 Miscellaneous Errors and Machine Errors

Error N°	Meaning of the error
1	Unknown character / Axis not recognised by the system
	Too many digits after a function
	Presence of a sign after a function which does not allow signs
	Non ISO character
	Sequence number over 32767
	Slash character "/" for block skipping is not the first in the block
	Spindle speed programming over > 65535
	Operation number negative or over 127
	Programming a null EF value
	Dwell value negative
	Minimum radius in tangential feed is negative
	Syntax error with L900-L925
	E.xxx programmable error number is null
	Unknown memory area in the syntax VAR H.. :s
	Executable block in a part program called on RAZ procedure (%110000)
	Programming of a rotary modulo axis in incremental mode (G91) with more than 15 revs
	Programming of a linear axis whose axis name is forbidden (depending on the current interpolation plane)
	Programming of feed F with a null value
	Programming of a spindle speed S over 65535
	Programming of S several times
	Programming of K several times
	Programming of M3/M4 several times
	Programming of {M40 or M41or ....or M45} several times
	Spindle already in movement
	G76+ H... ou G76- H.... program number must be the last data on the line
	G76+ N.. .... (block to insert) unknown name field to add
	G76+ N.. .... (block to insert) EA...EZ name field : block to insert is full
	G76+ N.. .... (block to insert) dimension over 100000000
	G76+ N.. .... (block to insert) error on a list of polynomial coefficients
	G76+ N.. .... (block to insert) block to insert is full
	G76+ H<bin> = H<source> .... Character '+' is missing after G76
	G76+ H<bin> = H<source> .... Reservation size is null
	G76+ H<bin> = H<source> .... Reserved size must be the last data on the line
	G76+ H<bin> = H<source> .... Unknown name field to add
	G76+ H<bin> = H<source> .... Syntax error with G76+
	Structured programming in polynomial programming
	The character after the operator @ (equivalent address) is not alphabetical
	Error on axis address value @
	The second character in a sequence Ex is not alphabetical
	Programming several times the retract axis
	Retract axis not existing
	Mirror function enabled on axis direction tool
2	Unknown G function or a mandatory argument missing after the G
3	Attribute of a G code wrongly positioned
4	Option not enabled or option parameter conflict
	Structured programming, RTCP, synchronised axes, etc.
5	«Geometric» option programming not enabled

6	«Polynomial interpolation» option missing coefficient table full
7	Error in programming movements parallel to inclined axes (grinder): - Programming is not in plane G20 - Interpolation is not in G00 or G01 - X is not programmed after G05 - X and Z are not programmed after G07
8	Tool correction number too high
9	A sequence of too many non-working blocks - Endless Loop
15	Invalid line configuration
16	Error in RTCP activation
17	End of block in a comment - close bracket missing
18 (1)	Servo error: P50 too small
20	No M02 at the end of the program
	Blocks not made executable in a cycle called by a G function
21	Blank definition incoherent in 3D mode
24	Error in inclined plane declaration
	Function reactivated when already active
	Function argument declaration incomplete
	Pivot point axis does not exist or is not servo-controlled
	Incoherent value in one of the matrix terms
25	Subroutine or sequence number does not exist
26	Too many subroutine nesting levels
27	Radius offset: In G52 machine origin programming / With taper threads
28	Syntax error in CCSPD or index table radius definition
	G96 must be followed by S / G97 must be followed by S / initial radius cannot be determined
	X or U not programmed in this block or a previous block
29	No range programmed for CCSPD / No range compatible with S in G97
	No range search option: S not included between min. and max. values of the range programmed
	With range search option: S does not belong to any range
30	Line error detected
32 (1)	Homing error / Axis already on limit switch
33 (1)	All slides on wait for synchronisation
34	Minimum radius reached in G21 interpolation
35 (1)	Sequence number not found in SEARCH
36 (1)	Part program memory full
37	Max. feed rate exceeded for thread cutting (COMAND)
38	Spindle already controlled by another channel
39 (1)	Axis synchronisation error (with axis synchronisation option)
40-49 (1)	Excessive following error on axis 0 to 9
50-59 (1)	Excessive following error on axis 10 to 19
60-69 (1)	Excessive following error on axis 20 to 29
70-71 (1)	Excessive following error on axis 30 and 31
72	Incremental programming after an incomplete block (PGP)
75	Switch from state G20 to G21 or G22: - last block in G20 incomplete as it is programmed in PGP or radius correction or with X ≤ 0 - first block in G21 without X and Y or G22 without Y and Z
	Switch from state G21 or G22 to G20: - last block in G21 or - G22 incomplete or - first block in G20 in mode G41 or - G42: In G21 or G22, initial radius negative or zero

76	In G21, programming of a fixed turning and milling cycle
77	Tool type incompatible with the machining phase (milling or turning)
78	Syntax error in programming channels synchronisation
	G78 P: Maximum 4 digits, must be less than the number of P must have a maximum of 4 digits and the index less than the number of channels
	G78 Q: Q must have a maximum of 4 digits and declared in ascending order
	No M00, M01 or M02 should be encountered when a G78 P... is active
(1) Machine error	

9.2 Parametric Programming Errors

Error N°	Meaning of the error
91	Parameter No. not recognised
92	Negative parameter assigned to a function which does not take a sign
	Parameter value higher than the maximum value of the function to which it is assigned
93	Error in parameter declaration or test expression
	L function not followed by symbols =,<,>, &, !
	Association with a prohibited function by a linking character -, +, *, /.
94	Operation prohibited in a parametric expression:
	Square root of a negative number / Division by 0
95	Attempt to write in an external input parameter or a read-only parameter
96	The block preceding the external parameter declaration is incomplete
	Programming of L100 ... in a contour definition in G64
97	Parameter update impossible in G76: - No symbol = after the parameter number - Less than 10 characters allocated for entry of the value
98	Write by a channel of a dynamic operation already used by another channel
99	Error related to the N/M AUTO function: - More than 5 N/M AUTO axes defined - Non-servo-controlled axis defined as N/M AUTO - Definition of an N/M AUTO axis of another channel

9.3 Profile Geometry Programming (PGP) Errors

9.3.1 The end point cannot be calculated from the elements in the blocks

Error N°	Meaning of the error
101	PGP: Insufficient data for programming a circle
	Circle programmed on two parallel axes (with R / see Error 107)
102	Line programmed by an angle and one coordinate with no way of calculating the other coordinate
106	In G02, G03, programming of the third axis without «helical» option
107	PGP: Circle programmed by its radius and end point, with the end point separated from the start point by more than 2 * radius
	Circle programmed by X, Z, I K with a start radius different from the end point (20 microns) / Helical: dimension of 3rd axis missing
	Circle programmed on two parallel axes (with I, J, K / see Error 101)

9.3.2 An intermediate point cannot be calculated from the data in two blocks

Error N°	Meaning of the error
110	PGP: Syntax error in the first of two blocks of a PGP entity
	Circle programmed on two parallel axes (with R / see Error 107)
111	PGP: Line/line intersection in which: - First block starting point = second block end point, or - First line angle = second line angle
113	PGP: The values programmed in the two blocks do not allow determination of an intersection or tangency point
114	PGP: Intersection or tangency point not determined by ET+, ET-, ES+ or ES-

9.3.3 An intermediate point cannot be calculated from the data in three blocks

Error N°	Meaning of the error
121	PGP: Syntax error in the last of the three blocks of a PGP entity
122	PGP: The first two blocks are non-intersecting lines
123	PGP: The data programmed in the three blocks do not allow determination of the tangent point
124	PGP: Tangent point of the second and third blocks not specified by ET+ or ET-

9.3.4 Fillet or chamfer definition errors

Error N°	Meaning of the error
130	Zero displacement in one of the two blocks connected by a fillet or a chamfer
131	Fillet or chamfer programmed in a block including M0, M1 or M2
	Programming insufficient in a sequence of blocks, not allowing determination of the end point
135	A chamfer can only connect two straight lines

9.3.5 Miscellaneous errors in PGP

Error N°	Meaning of the error
136	More than two blocks without movement between two geometric elements whose intersection or tangency point is to be calculated
137	Change of interpolation plane with an invalid block

9.4 Miscellaneous Errors

Error N°	Meaning of the error
138	Change of interpolation plane when not in G40 (FCU)
139	Two carried parallel axes programmed in the same block outside G52 and outside G00
140	Radius correction programming error: - Too many extraneous blocks between two consecutive paths
	The following functions cannot be programmed when radius offset is active: - M00, M01, M02, access to external parameters, writing of parameters E8xxxx or L > 100
141	Carried parallel axes: - Programming of a circle whose start point was programmed with one axis and whose end point was programmed with the associated parallel axis
143	Scale factor cancelled or enabled with radius offset
144	Movement of a quantified axis different from the increment
145	G29: ABS VAL (P * P + Q * Q + R * R - 1000 mm) > 1 mm (normal vector not a unit vector)
146	Offset in space / G29: - At least one of dimensions P, Q or R missing - At least one of dimensions X/U, Y/V or Z/W missing
148	Number of axes programmed exceeds the maximum authorised number
149	Tool radius too large with respect to programmed path

9.5 Request for Movements Outside the Machine Travel Limits

Error N°	Meaning of the error
150	Travel overrun on the X axis
151	Travel overrun on the Y axis
152	Travel overrun on the Z axis
153	Travel overrun on the U axis
154	Travel overrun on the V axis
155	Travel overrun on the W axis
156	Travel overrun on the A axis
157	Travel overrun on the B axis
158	Travel overrun on the C axis
159	Request for programmed movement on an UN-HOMED axis

9.6 Structured Programming Errors

Error N°	Meaning of the error
190	Too many branch or loop nesting levels (maximum 15)
191	Non-compliance with the syntax in structured programming structured programming prohibited in MDI mode the index of a FOR loop must be: - an L variable, a symbolic variable or a parameter E80000 - E81000 or E82000 non-compliance with the syntax of PUSH and PULL instructions - DO missing after WHILE programming of IF, THEN, ELSE in MDI mode
192	Keyword not recognised or prohibited in the context
193	Structure error
195	Program stack saturated / Too many constants defined for the space allocated
196	Error in array index declaration
197	Use of a symbol not declared as VAR
198	Syntax error in variable symbol declaration
199	Incorrect variable declaration syntax

9.7 Axis Errors

Error Nº	Meaning of the error
200 ÷ 203	Hardware error on handwheel 1 to 4
210 ÷ 219 (1)	Poor signal or sensor complementarity error on axis 0 to 9
220 ÷ 229 (1)	Poor signal or sensor complementarity error on axis 10 to 19
230 ÷ 239 (1)	Poor signal or sensor complementarity error on axis 20 to 29
240, 241 (1)	Poor signal or sensor complementarity error on axis 30 and 31
245 (1)	Fault on digital servo-control
248	DISC NT Drive in fault

(1) Machine error.

9.8 CNC - PLC Communication

Error Nº	Meaning of the error
249	Cyclic CNC - PLC communication fails

9.9 System options errors

Error Nº	Meaning of the error
250	Invalid PERSOChip
251	Inconsistent PERSOChip Data
252	Invalid Licence File
253	Nesting of subroutine called by M functions is forbidden

9.10 Axes Not Identified on the Bus

Error Nº	Meaning of the error
300 ÷ 309 (1)	Axis 0 to 9 declared in P2 but not detected on the bus
310 ÷ 319 (1)	Axis 10 to 19 declared in P2 but not detected on the bus
320 ÷ 329 (1)	Axis 20 to 29 declared in P2 but not detected on the bus
330, 331 (1)	Axis 30 and 31 declared in P2 but not detected on the bus

(1) Machine error. Caution: For this type of error a CNC reset causes a general reset (CNC reset + PLC reset).

9.11 Dynamic Operators in C

Error N°	Meaning of the error
400	Loading dynamic operators in C: The size of user code is too big
401	Loading dynamic operators in C: Format error
402	Loading dynamic operators in C: Checksum error
403	The system has insufficient memory for dynamic operators in C
404	Loading dynamic operators in C: Open error
405	Loading dynamic operators in C: Read error
406	Loading dynamic operators in C: Close error
407	Loading dynamic operators in C:The directory is empty
410	Dynamic operators in C: Number of parameters passed doesn't tally
411	Dynamic operators in C: USER ERROR from INIT: negative return
413	Unrecognised dynamic operators in C
414	Dynamic operators in C without MAIN
420	Dynamic operators in C: USER ERROR from the QUIT function
421	Dynamic operators in C: USER ERROR from the QUIT function: negative return
423	Dynamic operators in C: Range of function in C not from [0..100]

9.12 Spline Curve Interpolation Errors

Error N°	Meaning of the error
600	Curve number zero
601	N.. N.. must be programmed
602	No axes programmed in the first block of the contour
603	Curve slope undefined
604	Less than three blocks in the profile
605	Curve number unknown

9.13 Cycle Programming Errors

Error Nº	Meaning of the error
830	Positioning not completed
831	Spindle stopped
832	End point, P and K must be programmed
833	Retraction clearance too small
834	EB value: -90 < EB < +90
835	The values of P, Q, R and K are absolute values
836	The interpolation plane must be G81 or G20
837	Bad value of F or S
838	Threading resume : incoherent data
862	P or R and end point to be programmed
863	End point incoherent with EA
864	Milling tool prohibited in G66
871	Finished profile limits not defined
872	No dimensions in blank definition
873	P or R not programmed
874	Blank inconsistent with finished profile
875	No intersection of EA with the profile
876	Relief angle EB incorrectly defined
880	Cycle axis unknown
881	Parameter value not compatible
882	Hole bottom dimension not programmed
883	Pitch (I J K) or clearance (P) not programmed
884	More than 9 thread starts
885	Pocket incompatible with the plane selected
886	Tool incompatible with the radius programmed
887	Cut > tool diameter
888	Dwell prohibited in this cycle
889	Syntax error
890	Tool orientation incompatible
891	Return plane = bottom of hole
892	Axial feed missing
893	Lateral feed missing
894	ER prohibited in G20
895	G21, G22 prohibited in cycle
896	Dimension incompatible with tool radius
897	Length of oblong pocket < diameter
898	Tool corrector missing
899	Spindle not assigned to this channel or spindle or channel incompatible

Page deliberately empty

Symbol

! 456
@ 31
* 456, 481
/ 380, 456, 481
& 456, 481
% 45
+ 456, 481
< 457, 482
< = 457, 482
< > 457, 482
= 457, 482
> 457, 482
> = 457, 482
3D Curve 391
3D Curve Smoothing 391
3D tool correction
  with cylindrical tool 202
  with toroid or spherical tool 195
%9990 318
%11000 319
$ 397
$0 393
(PGP) Errors 541

A

Accurate stop at the end of a block 141
Addition 456, 481
Addresses
  characterising PGP 422
  without values 425
  with values 422
Addressing
  of a value 470
  of the PLC input terminals 489
  of the PLC output terminals 476, 488
Affaire number 476, 499
Angle element of a line 422
Angular offset 342
Arguments 49
  programmed alone 49
Arithmetic functions 456, 481
Arithmetic operations 456, 481
Assigning
  an external parameter to an NC function 483
  of a variable to an NC function 458
  of tool dimensions to the axes 414
Association of a slave axis with a master axis 524
Automatic homing 318, 401
Automatic homing subroutine branch 318
Auxiliary channels 409
Axes 23
  programmed by diameter 477, 512
Axes Not Identified 543
Axis
  acceleration anticipation time constant 478, 527
  acceleration at high speed 478, 526
  clamp 372
  coupling 477, 512
  declaration 315
  declared rotary modulo 360 degrees 523
  duplicated in manual mode (JOG) 525
  duplicated in the automatic modes 525
  measurement 508
  measurement correction (or spindle) 478, 525
  orientation 167
  origin switch 478, 522
  programming 55
  reference offset 478, 525
  retraction in the plane 241
  unclamp 372
Axis Errors 543

B

Backoff between two infeeds 227, 229
Basic Cycles 203
Bevel gear 169
Block
  deletion by G76- 326
  format 40
  insertion by G76+ 325
  processing 389
  sequencing without stopping movement 311
  skip 380
Blocks 43
Boring 208, 241
  and facing 241
  and Facing 231
  and facing cycle 231
  cycle 223
  cycle with dwell in hole bottom 233
  cycle with indexed spindle stop in hole bottom 225
Boring or mixed machines 412
Boring Type Machine 413
Boring type machines 412

C

Call 401
  by a Reset 401
  by M Function 402
  by the PLC 401
Calling
  a sequence of blocks with return 297
Call to subroutine return block 321
Cancel
  a canned cycle 205
  radius offset 176
Cancellation of MDI Modes 376
Canned cycles in the tool axis 203
carrier/carried axes 328
Cartesian coordinates 359
Centre drilling 206
Centre of rotation 29
Chamfer
  between two intersecting plines 424
  between two linear interpolations 139
Circular interpolation 114
  defined by three points 123
  plane selection and radius correction 91
Classification
  of miscellaneous M functions 50
  of preparatory G functions 48
Clearance after peck 241
Clockwise circular interpolation 114
Clockwise spindle rotation 92
CNC part programme 39
Codes C0 to C8 33
Common CNC Parameters 519
Common turning-milling functions 417
Comparison symbols for use with external parameters 482
Comparison symbols for use with variables 457
Concatenating blocks 389

Constant surface speed 97
Control of direction of rotation 92
Control of spindles 1 to 4 103
Conversion of the internal unit 457, 482
Coolant 373
1 373
2 373
Coolant off 373
coordinates 24
Coordinates
of the centre of a circle 423
of the end point of a circle. 422
of the end point of a line 422
Coordinate system 23
With C axis 357
Corrected length 34
Corrected radius 34
Cosine 456, 481
Counter boring 241
Counterclockwise circular interpolation 114
Creation of a programme 39
Creation of a programme by G76+ 322
Current channel number 499
Cutter tip radius 31
Cycle Programming Errors 545
Cylindrical coordinates 364

D

DAT1 27, 29
DAT2 27
DAT3 29, 328
Datum shift DAT1 and DAT2 enable 330
Datum shift DAT1 and DAT2 inhibit 330
Declaration of a variable in the programme 459
Decoded M functions 51
Definition
of a circle by its arc angle 130, 131
of an entity 421
of a spline curve 348
of geometric elements 427
of the addresses characterising the PGP 422
Of tool tip radius and orientation 33
Of travels and origins 25
Different from 457, 482
Discriminant 425
Division 456, 481, 530
DL 34
Double twist head 198
DR 34
Drilling Cycle with Chip Breaking 227
Drilling with chip breaking 241
Drip Feed Mode 320
Dwell 367
Dwell on each infeed 367
DX 35
Dynamic Operators in C 544
Dynamic tool correction 35
Dynamic tool correction definition 34
Dynamic tool length correction 34
Dynamic tool radius correction 34
DZ 35

E

EA 422
EB- 139
EB-.. 424
EB+ 138
EB+.. 423
E+/E- 425
EF 161
EG.. 381
Emergency retraction 315
Emergency Retraction 410
Emergency retract subroutine 315
Enabling and execution of functions stored in state G999 389
Encoded M functions 51
End
of hole 241
Entity 421
Equal to 457, 482
Equivalence of variables L900 to L925 461
Error messages 416
Errors 537
Machine Errors 537
Miscellaneous Errors 537, 542
ES 425
ET 425
Exchanging 411
Axes between Channels 411
Spindles Between Channels 411

F

Feed rate
in degrees per minute 148
in inches per revolution 155
in inverse time (V/D) 151
in millimetres per minute 149
in millimetres per revolution 155
limiting after interrupt 311
specific to chamfers EB- 161
specific to fillets EB+ 161
Feed rate limiting 361, 365
Fillet
between two interpolations 138
between two intersecting elements 423
Forced block sequencing 378
Front thread cutting 250
Functions 0 to 255 475, 498

G

General information on modes 21
General information on the system 21
General programme structure 45
Geometric elements 429
Geometric transformations
with 3-axis correction 199
with 5-axis correction 201
G function processing 389
G functions
incompatible with the programme state 48
with associated arguments 49
Greater than 457
Greater than or equal to 457
Groove roughing cycle 267

H

Helical interpolation 119
Homing
direction 478, 523
status on an axis 478, 521

status without switch wiring 478, 524
Homing Subroutine 401

I

I.. J.. 423
Inch data input 370
Inclined Axis 143
Inclined on a grinder
Inclined wheel 143
Infeed 206
Initial X value definition 358
ISO Functions 417
Milling cycles 417
Milling Function 417
Turning cycles 417
Turning Functions 417
ISO programme
structure 39
ISO programming 125

J

jerk 496
JOG 525

L

"L" 455
L + DL 34
Left radius correction 176
Less than 457, 482
Less than or equal to 457, 482
Linear interpolation 177
at programmed feed rate 191
List of
external parameters E 483
variables L 461
Local variables 455

M

Machine 23
axes 106
Machine coordinates 24
Machine Travel Limits 542
MDI Modes 376
Measured axis 523, 524, 526
Measurement
Origin 25
Origin offset 26
Measurement dimensions 28, 30
Meeting Point 407
Meeting point programming 407
Messages 351
Messages with parametric value 397
Message transmission to the display 393
M Functions 50
Milling functions 412
Mirror function 334
Miscellaneous M functions 48
Mixed machines 412
Modal G functions 48
Modal M functions 50
Mode 21
Movement origin selection 327
M post-functions 50
M pre-functions 50
Multiple Channels 401
Multiple start thread cutting 249
Multiplication 456, 481
MX 412

N

N/M-AUTO 478, 522
Nonmodal G functions 48
Nonmodal M functions 50
Normal vector 197
Notes on mixed machines (MX) 412
Notes related to machine axes 412
Number
of constant infeeds 211
of machine channels 499
of NC channels 476, 500
of total speed anticipation filter terms 475, 496
Numerical entry 395

O

Off-centring
of the table by DAT3 346
OP 27
Optional arguments 49
OR 456, 481
ORPOM 26
Other cycles 289
Other functions 367
Overspeed 369
Overspeed by handwheel 369

P

Parameters 401
External parameters E 401
Part origin offset 27
Peck drilling 210, 241
Peck drilling cycle 210
Plane selection 91
Polar angle reference axis 125
Polar coordinates 357
Polar Programming 125
of a circle 128
of a line 126
Positioning with stop at programmed distance 113
Preparatory G functions 48
Preselecting the programme origin 331, 416
probing 498
profile definition 444
Half part profile 445
Milled part #1 444
Milled Part#2 446
Turned part #1 448
Profile geometry programming 421
Programmable axes 106
Programme 21
analysis access parameters 491
creation by G76+ 322
deletion by G76- 324
dimensions 28, 30
numbering 47
origin 331
origin definition 413
origin offset 27, 328
Programme Variables 401
stop 374
variables L 455, 461

Programmed cancellation
of edit (EDIT) mode 376
Programmed feed stop 368
Programmed M function processing 389
Programming
in absolute or incremental dimensions 106
In cartesian coordinates 359
In cylindrical coordinates 364
in inches 148
in metric units 370
In polar coordinates 357
of A, B or C axes declared as non-rotary 59
of chamfers 138
of chamfers and fillets located between two elements 443
Of cycles 235
of entirely defined geometric elements 427
of fillets 138
of geometric elements 421
of geometric elements not entirely defined 429
of independent secondary axes 56
of movements 327
of PLC axes 410
of rotary axes modulo 360 degrees 57
of servoed rotary axes with limited excursion 59
Of variables L 468
With reference to diameter 88
With reference to radius 88
Programming examples
in PGP 444
of cycles G81-G89 242
of variables L 461
Of variables L 468
Programming notes
on variables L100-L199 and L900-L959 461

R

R.. 423
Radius correction 180
Radius of a circle 423
Rapid positioning 106
R + DR 34
Reaming 221, 223, 241
Reaming cycle 221, 223
Reset 319, 401
Restrictions due to drip feed mode 320
Retraction 241
Review 21
of axis orientation 23
of definitions 23
of machine 23
Right radius correction 176
Rigid tapping 217
Rigid Tapping Cycle 217
Rotary A, B and C axes 23, 59
Rotation centre offset 29
Rotation direction 92
Rough facing 259
Roughing 256
Roughing with groove 276
Rough turning 281
Rough turning/facing cycle 256

S

Scaling factor 382
enable 382
Secant element 425
Secondary linear U, V and W axes 24
Secondary U, V and W axes 55
Sending a message
to the display 393
Sequence 43
Sequence breaks 297
Sequence Interrupt 311
Sequences 313
Sequence without Return 305
Serial number 476, 499
Simple pocket cycle 289
Simple pockets 289
Sine 456, 481
software options bits 475, 498
Special Programming for Multiple Channels 400
Special programming of PLC axes 409
Speed of movement 106
Spindle 93
access parameters 521
and feed rate potentiometer enable 379
and feed rate potentiometer inhibiting 379
control 400
control selection 103
indexing 104
measurement 105
measurement selection 105
potentiometer 475, 492
Rotation speed (N) 96, 98
Speed 99
speed control 217
speed in rpm 94
Speed limiting 97, 99
speed range 102
stop 92
Spline 348
Spline curve 348
deletion 348, 356
execution command 348
interpolation 349
Spline interpolation 348
Square root 456, 481
Structured Programming Errors 542
subroutine %9990 318
subroutine %9990.i 401
Subroutine branch by M function 303
on reset 319
Subroutine branches 401
Subroutine branches and calls 46
Subroutine call by automatic control function 307
Subroutine display inhibited 47
Subroutine return block 321
Subroutines 47
Subtraction 456, 481
Surface speed (V) 97
Suspension of execution and forcing of block concatenation 389
Symbolic addressing
of Variables L900 to L925 461
of variables L926 to L951 461
System 21

T

Tangent element 425
Tangential feed rate 157
Tapping 214
Temporary suspension of next block preparation 313
Test on an external parameter for a conditional jump 485

Test on a variable for a conditional jump 460
T function 89
Thread chasing 238
Thread cutting
  By penetration to thread flank 248
  By straight penetration 248
  Without retraction slope 247
  With retraction slope 247
Tool
  correction call 170
  corrections 170, 171
  Cutting part radius 33
  dimensions 31
  position with respect to the part 176
  programming 178
  radius 31
  radius “R” of correction xxx 476, 503
  retractions 212
  Retraction slope 247
  type of correction xxx 476, 504
  type selection 414
Tool correction 365
Tool corrections 361
Tool dimensions corrections 172
Tool disengagement 184
Tool inside the profile 183, 184
Tool length correction 171
Tool outside the profile 181, 183
Tool positioning 179
Tool radius corrections 172
Tool tip
  Orientation 33
  Orientation code 33
Toroidal and Spherical Tools 198
Transferring the current values of parameters E 398
Transferring the current values of variables L 398
Travels 25
Truncating 456
Turn/face roughing cycle 256
Turning functions 412

U

Unconditional subroutine branch 297

V

Variable Assignments 455
Variable L programming syntax 458
Variables
  L100 to L199 455
  L900 to L925 461
  L900 to L959 455
  L926 to L951 461
'virtual' axis 494

W

Word 40
  defining a dimension 42
  format 43
  not defining a dimension 42
Writing a block defining
  a path 44
  a tool change 43
  the start of rotation 44
Writing a programme 22

X

X and U axes with reference to diameter or radius 412
X.. Y.. 422
XY plane 91

Y

YZ plane 91

Z

Zero point 21, 23
ZX plane 417

Page deliberately empty

Published by NUM Spa

Via F. Somma 62
20012 Cuggiono (MI) (Italy)
Tel. : +39 02 979 691
website: www.num.com

Printed February 2013

CNC HighEnd Applications