



编程基础知识

1

三种模式的 G 代码

2

运行指令

3

测量指令

4

其它功能

5

SINUMERIK

SINUMERIK 808D

车削 第三部分：编程  
(ISO 编程指令)

编程和操作手册

适用于： SINUMERIK 808D 车削（软件版本： V4.4.2）

目标使用人群： 最终用户及服务工程师

12/2012

6FC5398-5DP10-0RA0

## 法律资讯

### 警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。



表示如果不采取相应的小心措施，**将会**导致死亡或者严重的人身伤害。



表示如果不采取相应的小心措施，**可能**导致死亡或者严重的人身伤害。



表示如果不采取相应的小心措施，可能导致轻微的人身伤害。



表示如果不采取相应的小心措施，可能导致财产损失。

当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

### 合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

### 按规定使用 Siemens 产品

请注意下列说明：



**Siemens** 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 **Siemens** 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

### 商标

所有带有标记符号 ® 的都是西门子股份有限公司的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

### 责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

# 目录

<b>1 编程基础知识 .....</b>	<b>7</b>
1.1 引言 .....	7
1.1.1 西门子模式 .....	7
1.1.2 ISO 编程指令模式 .....	7
1.1.3 切换运行方式 .....	7
1.1.4 显示 G 代码 .....	8
1.1.5 最大轴/轴标识符数量 .....	9
1.1.6 定义 G 代码系统 A, B 或 C .....	9
1.1.7 小数点的编程 .....	10
1.1.8 注释 .....	12
1.1.9 Skip block (跳过程序段) .....	12
1.2 进给率的前提条件 .....	13
1.2.1 快速移动 .....	13
1.2.2 轨迹进给率 (F 功能) .....	13
1.2.3 线性进给率(G98) .....	15
1.2.4 旋转进给率(G99) .....	15
<b>2 三种模式的 G 代码 .....</b>	<b>17</b>
2.1 SINUMERIK ISO 编程指令车削版模式 A .....	17
2.2 SINUMERIK ISO 编程指令车削版模式 B .....	19
2.3 SINUMERIK ISO 编程指令车削版模式 C .....	23
<b>3 运行指令 .....</b>	<b>27</b>
3.1 插补指令 .....	27
3.1.1 快速运行(G00) .....	27
3.1.2 直线插补 (G01) .....	29
3.1.3 圆弧插补(G02, G03) .....	30
3.1.4 轮廓段编程以及插入倒角和倒圆 .....	34
3.2 使用 G 功能回参考点 .....	36
3.2.1 通过中间点回参考点(G28) .....	36
3.2.2 参考位置点检查(G27) .....	37
3.2.3 通过参考点选择回参考点(G30) .....	37
3.3 切削螺纹功能的应用 .....	38
3.3.1 等螺距螺纹切削 (G32) .....	38
3.3.2 螺纹链接 (G32) .....	40
3.3.3 多头螺纹切削 (G32) .....	41
3.3.4 可变螺距螺纹切削 (G34) .....	43

---

<b>4 测量指令 .....</b>	<b>45</b>
4.1 坐标系 .....	45
4.1.1 机床坐标系(G53) .....	46
4.1.2 工件坐标系(G50) .....	46
4.1.3 复位刀具坐标系(G50.3) .....	47
4.1.4 选择工件坐标系 .....	47
4.1.5 写入零点偏移/刀具补偿(G10) .....	48
4.2 确定坐标值的输入方式 .....	50
4.2.1 X 轴的直径和半径编程 .....	50
4.2.2 英制/公制尺寸输入(G20, G21) .....	51
4.3 控制时间的指令 .....	52
4.4 刀具补偿功能 .....	53
4.4.1 刀具补偿数据存储器 .....	53
4.4.2 刀具长度补偿 .....	54
4.4.3 刀沿半径补偿 (G40, G41/G42) .....	55
4.5 S 功能, T 功能, M 功能和 B 功能 .....	60
4.5.1 主轴功能 (S 功能) .....	60
4.5.2 恒定切削速度 (G96, G97) .....	60
4.5.3 用 T 指令 (T 功能) 换刀 .....	62
4.5.4 附加功能 (M 功能) .....	62
4.5.5 用于控制主轴的 M 功能 .....	63
4.5.6 用于调用子程序的 M 功能 .....	64
4.5.7 通过 M 功能调用宏 .....	64
4.5.8 M 功能 .....	66
<b>5 其它功能 .....</b>	<b>67</b>
5.1 程序支持功能 .....	67
5.1.1 固定循环 .....	67
5.1.2 多次重复循环 .....	76
5.1.3 钻孔循环 (G80 至 G89) .....	92
5.2 可编程数据输入 .....	103
5.2.1 更改刀具补偿值 (G10) .....	103
5.2.2 用于调用子程序的 M 功能(M98, M99) .....	104
5.3 八位数的程序号 .....	105
5.4 测量功能 .....	106
5.4.1 通过 G10.6 快速退刀 .....	106
5.4.2 删剩余行程的测量 (G31) .....	107
5.4.3 通过 G31, P1 - P4 测量 .....	108
5.4.4 通过 M96/M97 (ASUP) 编程中断程序 .....	109
5.5 宏程序 .....	111
5.5.1 与子程序的区别 .....	111

---

5.5.2	调用宏程序(G65, G66, G67).....	111
5.6	附加功能.....	119
5.6.1	G05 .....	119
5.6.2	多边形车削 .....	120
5.6.3	ISO 编程指令模式中的压缩程序 .....	121
5.6.4	DryRun 和跳转级的切换模式.....	122
5.6.5	通过 M96, M97 写入中断程序.....	123
	索引 .....	125



# 编程基础知识

## 1.1 引言

### 1.1.1 西门子模式

在生效的西门子模式中以下条件有效：

- 在每个通道中都可以通过机床数据 20150 \$MC\_GCODE\_RESET\_VALUES 定义 G 指令的缺省设置。
- 西门子编程指令模式下不可写入任何 ISO 编程指令。

### 1.1.2 ISO 编程指令模式

在生效的 ISO 编程指令模式中以下条件生效：

- 可以通过机床数据将 ISO 编程指令模式设为数控系统的缺省设置。缺省情况下数控系统随后在 ISO 编程指令模式中启动。
- 在 ISO 模式下只能写入由 ISO 编程指令组成的 G 功能，而不能写入西门子 G 功能。
- 在同一个 NC 程序段中不允许混用 ISO 编程指令和西门子编程指令。
- 也无法借助一个 G 指令切换“ISO 编程指令 M”和“ISO 编程指令 T”。
- 但在该模式下可以调用西门子模式下编写的子程序。
- 如果需要使用西门子功能，必须首先切换到西门子模式。

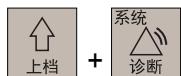
### 1.1.3 切换运行方式

SINUMERIK 808D 支持以下两种编程指令模式：

- 西门子编程指令模式
- ISO 编程指令模式

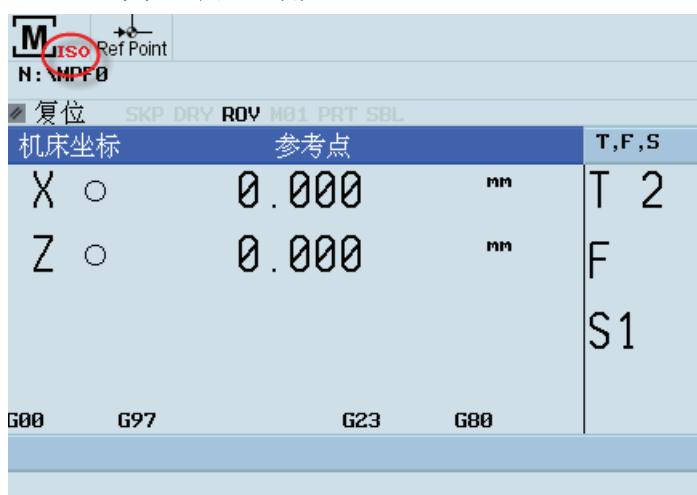
须注意模式切换不会影响生效的刀具、刀具补偿和零点偏移。

## 步骤



**ISO模式**

1. 选择相关操作区域并进入其主画面。
2. 在垂直软键栏上按下该软键。数控系统自动开始从西门子模式切换至 ISO 编程指令模式。ISO 编程指令模式生效后，图标 "ISO" 显示在画面左上角。



如需从 ISO 模式切回西门子模式，可再次按下该软键。

### 1.1.4 显示 G 代码

显示 G 代码的编程指令和当前程序段的编程指令相同，即西门子编程指令或 ISO 编程指令。如果通过 DISPLOF 取消程序段的显示，则继续以之前显示生效程序段的编程指令显示 G 代码。

## 示例

使用 ISO 编程指令模式的 G 功能调用西门子标准循环。为此，在相应循环的开头写入 DISPLOF，从而可以在显示中保留 ISO 编程指令中写入的 G 功能。

```
PROC CYCLE328 SAVE DISPLOF
N10 ...
...
N99 RET
```

## 步骤

通过主程序调用西门子 shell 循环。调用该循环后自动选择西门子模式。

使用 DISPLOF 抑制调用循环时的程序段显示，而继续在 ISO 模式中显示 G 代码。

在循环结束时，通过属性“SAVE”将 shell 循环中修改的 G 代码恢复为初始状态。

### 1.1.5 最大轴/轴标识符数量

ISO 编程指令模式下可以最多写入 9 个轴。前三个轴的标识符固定为 X、Y、Z。而所有其它轴可以用字母 A、B、C、U、V、W 命名。

### 1.1.6 定义 G 代码系统 A, B 或 C

在 ISO-语言-车削版中，G 代码系统 A, B 或 C 是不同的。默认 G 代码系统 A 有效。通过 MD10881 \$MN\_MM\_EXTERN\_GCODE\_SYSTEM 对 G 代码系统 A, B 或 C 进行如下选择：

\$MN\_MM\_EXTERN\_GCODE\_SYSTEM = 0: G 代码系统 B

\$MN\_MM\_EXTERN\_GCODE\_SYSTEM = 1: G 代码系统 A

\$MN\_MM\_EXTERN\_GCODE\_SYSTEM = 2: G 代码系统 C

#### G 代码系统 A

如果 G 代码系统 A 有效，则 G91 不可用。这时要对轴 X, Y 和 Z 使用地址符 U, V 和 W 编程增量轴运行。地址符 U, V 和 W 在这种情况下不能作为轴名称使用，因为会导致轴的最大数量减少到 6 个。

在 G 代码系统 A 中地址 H 用于编程 C 轴的增量运行。

为了使 shell 循环在正确的 G 代码系统中运行，必须将相应的系统输入到 GUD 变量 \_ZSFI[39] 中。

---

#### 说明

如未作其他说明，本资料中都是以 G 代码系统 A 为例。

---

### 1.1.7 小数点的编程

对于编程不带小数点的值，ISO 编程指令模式中提供两种记数法：

- **计算器记数法**

不带小数点的数值会视为毫米、英寸或度。

- **标准记数法**

不带小数点的数值和一个换算系数相乘。

通过 MD10884 EXTERN\_FLOATINGPOINT\_PROG 进行设置。

存在两种不同的换算系数：IS-B 和 IS-C。赋值以地址 X Y Z U V W A B C I J K Q R 和 F 为基准。

通过 MD10886 EXTERN\_INCREMENT\_SYSTEM 进行设置。

示例：

线性轴，毫米：

- X 100.5

相当于带小数点的数值：100.5 mm

- X 1000

– 计算器记数法：1,000 mm

– 标准记数法：

IS-B:  $1000 * 0.001 = 1 \text{ mm}$

IS-C:  $1000 * 0.0001 = 0.1 \text{ mm}$

## ISO 编程指令车削

表格 1- 1 不同的换算系数：IS-B 和 IS-C

地址	单位	IS-B	IS-C
线性轴	毫米	0,001	0,0001
	英制	0,0001	0,00001
回转轴	度	0,001	0,0001
F 进给率 G94 (毫米每分钟/英寸每分钟)	毫米	1	1
	英制	0,01	0,01

地址	单位	IS-B	IS-C
F 进给率 G95 (毫米每转/英寸每转) \$MC_EXTERN_FUNCTION_MASK			
位 8 = 0	毫米 英制	0,01 0,0001	0,01 0,0001
位 8 = 1	毫米 英制	0,0001 0,000001	0,0001 0,000001
F 螺距	毫米 英制	0,0001 0,000001	0,0001 0,000001
C 倒角	毫米 英制	0,001 0,001	0,0001 0,0001
R 半径 G10 toolcorr	毫米 英制	0,001 0,001	0,0001 0,0001
I、J、K IPO 参数	毫米 英制	0,001 0,001	0,0001 0,0001
G04 X 或者 U		0,001	0,001
A 轮廓段角度		0,001	0,001
G76,G92 攻丝循环 \$MC_EXTERN_FUNCTION_MASK 位 8 = 0 F 作为进给率, 同 G94、G95 位 8 = 1 F 作为螺距			
G84,G88 攻丝循环 \$MC_EXTERN_FUNCTION_MASK			
位 9 = 0 G95 F	毫米 英制	0,01 0,0001	0,01 0,0001
位 8 = 1 G95 F	毫米 英制	0,0001 0,000001	0,0001 0,000001

### 1.1.8 注释

在 ISO 编程指令模式中，圆括号被视为注释符。而在西门子模式中，“;”被视为注释符。为简化编程，在 ISO 编程指令模式中“;”同样被视为注释符。

如果在一个注释中再次出现左括号“（”，则再次闭合所有括号后，注释才结束。

示例：

```
N5 (注释) X100 Y100  
N10 (注释(注释)) X100 Y100  
N15 (注释(注释) X100) Y100
```

在程序段 N5 和 N10 中执行 X100 Y100；在程序段 N15 中只执行 Y100，因为在 X100 后才闭合第一个括号。在它之前的指令都视为注释。

### 1.1.9 Skip block (跳过程序段)

跳转或抑制程序段的符号“/”可以出现在程序段的任意位置，也就是说可以位于程序段中段。如果在编译时写入的程序段跳转级生效，则从该位置起停止编译，直至程序段结束。因此，一个生效的程序段跳转级也就相当于一个程序段结尾。

示例：

```
N5 G00 X100. /3 YY100 --> 报警 12080 “语法错误”  
N5 G00 X100. /3 YY100 --> 没有报警，因为程序跳转级 3 生效
```

注释内的程序段跳转符不视为程序段跳转符

示例：

```
N5 G00 X100. (/3 工件 1) Y100  
; 即使程序段跳转级 3 生效时也运行 Y 轴
```

可以激活程序段跳转级 /1 到 /9。<1 和 >9 的程序段跳转级会导致报警 14060“非法跳转级，程序段跳转出错”。

该功能相当于当前的西门子跳转级。和 ISO 原始编程指令相反，必须单独激活“/”和“/1”分隔的跳转级。

---

#### 说明

“/0”时可以省略值“0”。

---

## 1.2 进给率的前提条件

下文为您介绍了用于确定切削刀具进给速度（每分钟或每转的位移）的进给功能。

### 1.2.1 快速移动

快速移动不仅可以用于定位(G00)，而且可以用于快速手动运行(JOG)。在快速移动中，每个轴按照各自设定的快速移动速度运行。机床制造商可通过机床数据确定各个轴的快速移动速度。轴独立运行，所以每个轴在不同时间到达目标位置。因此，形成的刀具轨迹一般不是直线。

### 1.2.2 轨迹进给率 (F 功能)

#### 说明

如果没有特殊说明，本资料中给出的切削刀具进给速度始终为“毫米/分钟”。

地址符“F”表示线性插补(G01)或圆弧插补(G02, G03)时刀具的进给率。

地址符“F”后输入切削刀具的进给率，单位“毫米/分钟”。

F 的取值范围参见机床制造商的说明资料。

最大进给率可能受伺服系统和机械构造的限制。通过机床数据设定最大进给率，当前进给率始终限制在该值范围内。

通常情况下，轨迹进给率由所有参与运动的几何轴的各个速度分量组成；并且以刀具中心点为参照（参见下图）。

## 1.2 进给率的前提条件

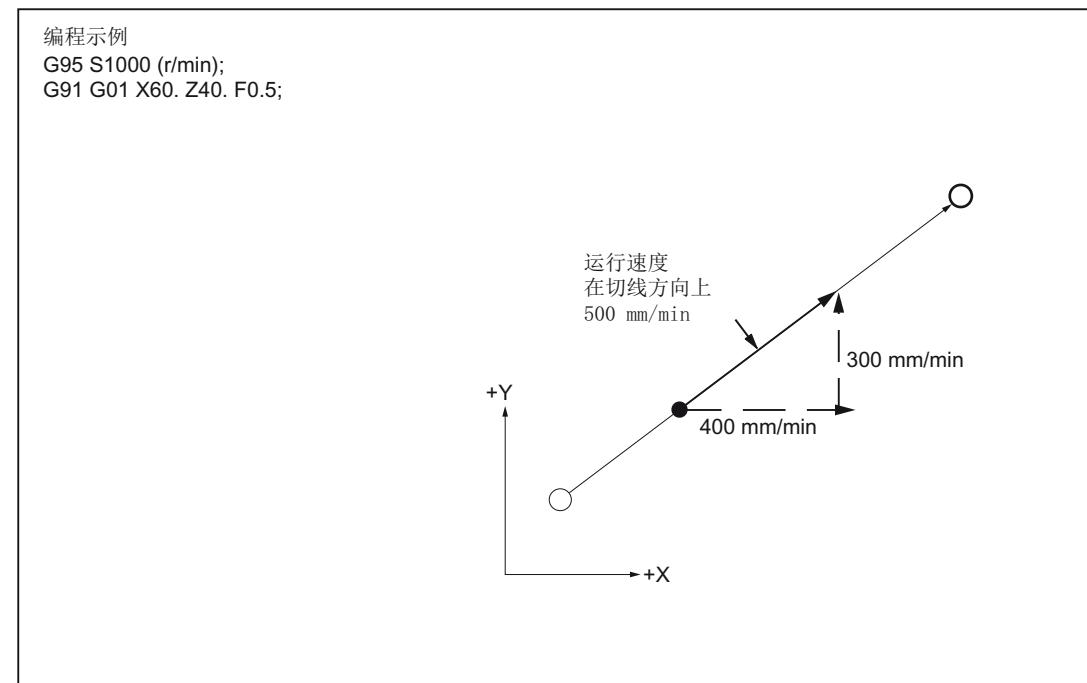


图 1-1 2 个轴的线性插补

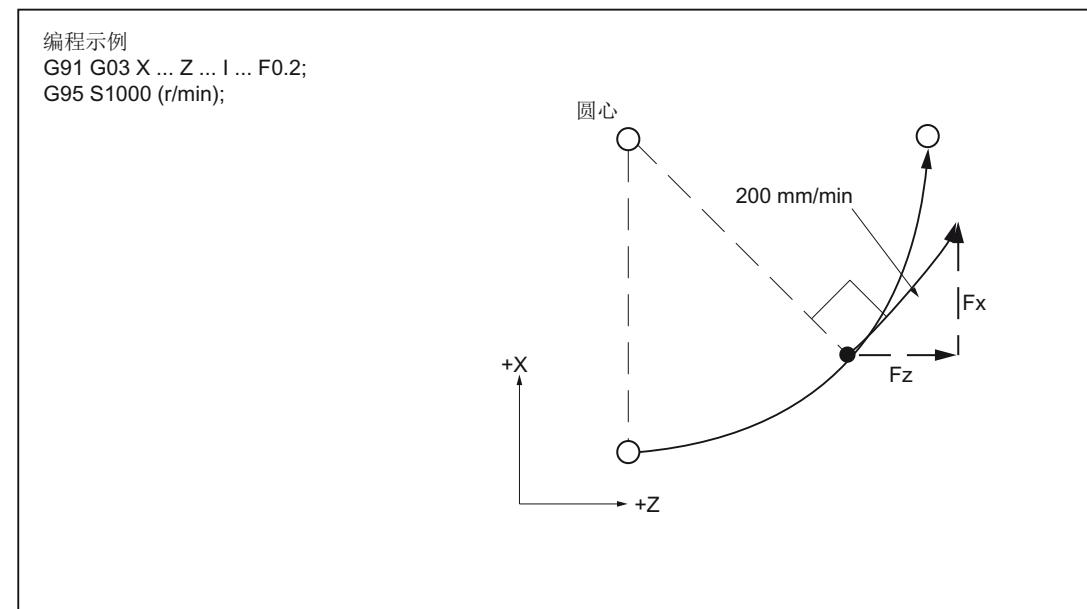


图 1-2 2 个轴的圆弧插补

### 说明

如果写入了“F0”并且功能“固定进给率”在程序段中没有激活，则输出报警 14800“通道 %1 程序段 %2 编程的轨迹速度小于或等于零”。

### 1.2.3 线性进给率(G98)

给定 G98 后，执行地址符 F 后给出的进给率，单位：毫米/分钟、英寸/分钟或度/分钟。

### 1.2.4 旋转进给率(G99)

写入 G99 后执行适用于主主轴的进给率，单位：毫米/转或英寸/转。

#### 说明

所有的指令模态有效。如果需要在 G98 与 G99 之间切换 G 指令进给率，则应重新写入轨迹进给值。在用回转轴加工时，进给率也可以用度/转来表示。

## 1.2 进给率的前提条件

## 三种模式的 G 代码

### 2.1

### SINUMERIK ISO 编程指令车削版模式 A

表格 2- 1 SINUMERIK ISO 编程指令车削版 - 模式 A

名称	索引	说明	格式
<b>01. G 代码组 (模态)</b>			
G0	1	快速移动	G00 X... Y... Z... ;
G1	2	直线运行	G01 X... Z... F... ;
G2	3	顺时针圆弧/螺线	G02(G03) X(U)... Z(W)... I... K... (R...) F... ;
G3	4	逆时针圆弧/螺线	
G32	5	等螺距螺纹切削	G32 X (U)... Z (W)... F... ;
G90	6	轴向切削循环	G... X... Z... F... ;
G92	7	螺纹切削循环	G... X... Z... F... Q... ;
G94	8	径向切削循环	G... X... Z... F... ;
G34	9	可变螺距螺纹切削	G34 X (U)... Z (W)... F... K... ;
<b>02. G 代码组 (模态)</b>			
G96	1	恒定切削速度“开”	G96 S...
G97	2	恒定切削速度“关”	G97 S...
<b>04. G 代码组 (模态)</b>			
G68	1	双转塔/刀架“开”	
G69	2	双转塔/刀架“关”	
<b>06. G 代码组 (模态)</b>			
G20	1	英制输入系统	
G21	2	公制输入系统	
<b>07. G 代码组 (模态)</b>			
G40	1	取消铣刀半径补偿	
G41	2	轮廓左侧补偿	
G42	3	轮廓右侧补偿	
<b>08. G 代码组 (模态)</b>			
<b>10. G 代码组 (模态)</b>			
G80	1	钻孔循环“关”	G80;
G83	2	端面深孔钻削	G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... ;
G84	3	端面螺纹钻削	G84 X(U)... C(H)... Z(W)... R... P... F... M... K... ;

## 三种模式的 G 代码

### 2.1 SINUMERIK ISO 编程指令车削版模式 A

名称	索引	说明	格式
G85	4	端面钻孔循环	G85 X(U)... C(H)... Z(W)... R... P... F... K... M... ;
G87	5	侧面深孔钻削	G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... ;
G88	6	侧面螺纹钻削	G88 Z(W)... C(H)... X(U)... R... P... F... M... K... ;
G89	7	侧面钻孔	G89 Z(W)... C(H)... X(U)... R... P... F... K... M... ;
12. G 代码组（模态）			
G66	1	宏模态调用	G66 P... L... <参数>;
G67	2	删除宏模态调用	G67 P... L... <参数>;
14. G 代码组（模态）			
G54	1	选择零点偏移	
G55	2	选择零点偏移	
G56	3	选择零点偏移	
G57	4	选择零点偏移	
G58	5	选择零点偏移	
G59	6	选择零点偏移	
16. G 代码组（模态）			
G17	1	XY 平面	
G18	2	ZX 平面	
G19	3	YZ 平面	
18. G 代码组（非模态）			
G4	1	暂停 [s] 或主轴旋转	G04 X...; 或 G04 P...;
G10	2	写入零点偏移/刀具补偿	G10 L2 Pp X... Z... ;
G28	3	1. 返回参考点	G28 X... Z... ;
G30	4	2./3./4. 返回参考点	G30 Pn X... Z... ;
G31	5	删除剩余行程的测量	G31 X... Y... Z... F_;
G52	6	可编程的零点偏移	
G65	7	宏调用	G65 P_ L_ ;
G70	8	精加工循环	G70 P... Q... ;
G71	9	轴向粗车循环	G71 U... R... ;
G72	10	径向粗车循环	G72 W... R... ;
G73	11	封闭切削循环	G73 U... W... R... ;
G74	12	轴向切槽多重循环	G74 R... ;
G75	13	径向切槽多重循环	G75 R... ; 或 G75 X(U)... Z(W)... P... Q... R... F... ;
G76	14	多重螺纹切削循环	G76 P... (m, r, a) Q... R... ;
G50	15	设置实际值	G92 (G50) X... Z... ;
G27	16	回参考点运行检查（准备就绪）	G27 X... Z... ;

名称	索引	说明	格式
G53	17	返回机床坐标系中的位置	G53 X... Z... ;
G10.6	19	快速提刀打开/关闭	
G5	20	高速循环切削	G05 Pxxxxx Lxxx ;
G30.1	21	参考点位置	-
G5.1	22	高速循环 (High-speed cycle) -> 调用 CYCLE305	-
G50.3	23	删除实际值, 复位 WCS	
G60	24	定向定位	
20. G 代码组 (模态)			
G50.2	1	多边形车削“关”	
G51.2	2	多边形车削“开”	G51.2 P...Q...;
31. G 代码组 (模态)			
G290	1	选择西门子模式	-
G291	2	选择 ISO 编程指令模式	-

## 2.2 SINUMERIK ISO 编程指令车削版模式 B

表格 2-2 SINUMERIK ISO 编程指令车削版 - 模式 B

名称	索引	说明	格式
01. G 代码组 (模态)			
G0	1	快速移动	G00 X... Y... Z... ;
G1	2	直线运行	G01 X... Z... F... ;
G2	3	顺时针圆弧/螺线	G02(G03) X(U)... Z(W)... I... K... (R...) F... ;
G3	4	逆时针圆弧/螺线	
G33	5	恒螺距螺纹切削	G33 X (U)... Z (W)... F... ;
G77	6	轴向切削循环	G.. X... Z... F... ;
G78	7	螺纹切削循环	G... X... Z... F... Q... ;
G79	8	径向切削循环	G... X... Z... F... ;
G34	9	可变螺距螺纹切削	G34 X (U)... Z (W)... F... K... ;
02. G 代码组 (模态)			
G96	1	恒定切削速度“开”	G96 S...
G97	2	恒定切削速度“关”	G97 S...

## 三种模式的 *G* 代码

### 2.2 SINUMERIK ISO 编程指令车削版模式 B

名称	索引	说明	格式
03. G 代码组（模态）			
G90	1	绝对编程	
G91	2	增量编程	
04. G 代码组（模态）			
G68	1	双转塔/刀架“开”	
G69	2	双转塔/刀架“关”	
05. G 代码组（模态）			
G94	1	线性进给率, [毫米/分, 英寸/分]	
G95	2	旋转进给率, [毫米/转, 英寸/转]	
06. G 代码组（模态）			
G20	1	英制输入系统	
G21	2	公制输入系统	
07. G 代码组（模态）			
G40	1	取消铣刀半径补偿	
G41	2	轮廓左侧补偿	
G42	3	轮廓右侧补偿	
08. G 代码组（模态）			
10. G 代码组（模态）			
G80	1	钻孔循环“关”	G80;
G83	2	端面深孔钻削	G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... ;
G84	3	端面螺纹钻削	G84 X(U)... C(H)... Z(W)... R... P... F... M... K... ;
G85	4	端面钻孔循环	G85 X(U)... C(H)... Z(W)... R... P... F... K... M... ;
G87	5	侧面深孔钻削	G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... ;
G88	6	侧面螺纹钻削	G88 Z(W)... C(H)... X(U)... R... P... F... M... K... ;

名称	索引	说明	格式
G89	7	侧面钻孔	G89 Z(W)... C(H)... X(U)... R... P... F... K... M... ;
<b>11. G 代码组 (模态)</b>			
G98	1	钻孔循环时返回出发点	
G99	2	钻孔循环时返回点 R	
<b>12. G 代码组 (模态)</b>			
G66	1	宏模态调用	G66 P... L... <参数>;
G67	2	删除宏模态调用	G67 P... L... <参数>;
<b>14. G 代码组 (模态)</b>			
G54	1	选择零点偏移	
G55	2	选择零点偏移	
G56	3	选择零点偏移	
G57	4	选择零点偏移	
G58	5	选择零点偏移	
G59	6	选择零点偏移	
<b>16. G 代码组 (模态)</b>			
G17	1	XY 平面	
G18	2	ZX 平面	
G19	3	YZ 平面	
<b>18. G 代码组 (非模态)</b>			
G4	1	暂停 [s] 或主轴旋转	G04 X...; 或 G04 P...;
G10	2	写入零点偏移/刀具补偿	G10 L2 Pp X... Z... ;
G28	3	1. 返回参考点	G28 X... Z... ;
G30	4	2./3./4. 返回参考点	G30 Pn X... Z... ;
G31	5	删除剩余行程的测量	G31 X... Y... Z... F_;
G52	6	可编程的零点偏移	
G65	7	宏调用	G65 P_ L_ ;
G70	8	精加工循环	G70 P... Q... ;
G71	9	轴向粗车循环	G71 U... R... ;

## 三种模式的 *G* 代码

### 2.2 SINUMERIK ISO 编程指令车削版模式 B

名称	索引	说明	格式
G72	10	径向粗车循环	G72 W... R... ;
G73	11	封闭切削循环	G73 U... W... R... ;
G74	12	轴向切槽多重循环	G74 R... ;
G75	13	径向切槽多重循环	G75 R... ; 或 G75 X(U)... Z(W)... P... Q... R... F... ;
G76	14	多重螺纹切削循环	G76 P... (m, r, a) Q... R... ;
G92	15	设置实际值	G92 (G50) X... Z... ;
G27	16	回参考点运行检查 (准备就绪)	G27 X... Z... ;
G53	17	返回机床坐标系中的位置	(G90) G53 X... Z... ;
G10.6	18	快速提刀打开/关闭	
G5	19	高速循环切削	G05 Pxxxxx Lxxx ;
G30.1	20	参考点位置	
G5.1	21	高速循环 (High-speed cycle) -> 调用 CYCLE305	
G92.1	22	删除实际值, 复位 WCS	
G60	23	定向定位	
20. G 代码组 (模态)			
G50.2	1	多边形车削“关”	
G51.2	2	多边形车削“开”	G51.2 P...Q...;
31. G 代码组 (模态)			
G290	1	选择西门子模式	
G291	2	选择 ISO 编程指令模式	

## 2.3

## SINUMERIK ISO 编程指令车削版模式 C

表格 2-3 SINUMERIK ISO 编程指令车削版 - 模式 C

名称	索引	说明	格式
<b>01. G 代码组 (模态)</b>			
G0	1	快速移动	G00 X... Y... Z... ;
G1	2	直线运行	G01 X... Z... F... ;
G2	3	顺时针圆弧/螺线	G02(G03) X(U)... Z(W)... I...
G3	4	逆时针圆弧/螺线	K... (R...) F... ;
G33	5	恒螺距螺纹切削	G33 X (U)... Z (W)... F... ;
G20	6	轴向切削循环	G... X... Z... R... F... ;
G21	7	螺纹切削循环	G... X... Z... F... Q... ;
G24	8	径向切削循环	G... X... Z... F... ;
G34	9	可变螺距螺纹切削	G34 X (U)... Z (W)... F... K... ;
<b>02. G 代码组 (模态)</b>			
G96	1	恒定切削速度“开”	G96 S...
G97	2	恒定切削速度“关”	G97 S...
<b>03. G 代码组 (模态)</b>			
G90	1	绝对编程	
G91	2	增量编程	
<b>04. G 代码组 (模态)</b>			
G68	1	双转塔/刀架“开”	
G69	2	双转塔/刀架“关”	
<b>05. G 代码组 (模态)</b>			
G94	1	线性进给率, [毫米/分, 英寸/分]	
G95	2	旋转进给率, [毫米/转, 英寸/转]	
<b>06. G 代码组 (模态)</b>			
G70	1	英制输入系统	G70 P... Q... ;
G71	2	公制输入系统	G71 U... R... ;

名称	索引	说明	格式
<b>07. G 代码组 (模态)</b>			
G40	1	取消铣刀半径补偿	
G41	2	轮廓左侧补偿	
G42	3	轮廓右侧补偿	
<b>08. G 代码组 (模态)</b>			
<b>10. G 代码组 (模态)</b>			
G80	1	钻孔循环“关”	G80;
G83	2	端面深孔钻削	G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... ;
G84	3	端面螺纹钻削	G84 X(U)... C(H)... Z(W)... R... P... F... M... K... ;
G85	4	端面钻孔循环	G85 X(U)... C(H)... Z(W)... R... P... F... K... M... ;
G87	5	侧面深孔钻削	G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... ;
G88	6	侧面螺纹钻削	G88 Z(W)... C(H)... X(U)... R... P... F... M... K... ;
G89	7	侧面钻孔	G89 Z(W)... C(H)... X(U)... R... P... F... K... M... ;
<b>11. G 代码组 (模态)</b>			
G98	1	钻孔循环时返回出发点	
G99	2	钻孔循环时返回点 R	
<b>12. G 代码组 (模态)</b>			
G66	1	宏模态调用	G66 P... L... <参数>;
G67	2	删除宏模态调用	G67 P... L... <参数>;
<b>14. G 代码组 (模态)</b>			
G54	1	选择零点偏移	
G55	2	选择零点偏移	
G56	3	选择零点偏移	
G57	4	选择零点偏移	

名称	索引	说明	格式
G58	5	选择零点偏移	
G59	6	选择零点偏移	
<b>16. G 代码组 (模态)</b>			
G17	1	XY 平面	
G18	2	ZX 平面	
G19	3	YZ 平面	
<b>18. G 代码组 (非模态)</b>			
G4	1	暂停 [s] 或主轴旋转	G04 X...; 或 G04 P...;
G10	2	写入零点偏移/刀具补偿	G10 L2 Pp X... Z... ;
G28	3	1. 返回参考点	G28 X... Z... ;
G30	4	2./3./4. 返回参考点	G30 Pn X... Z... ;
G31	5	删除剩余行程的测量	G31 X... Y... Z... F_;
G52	6	可编程的零点偏移	
G65	7	宏调用	G65 P_ L_ ;
G72	8	精加工循环	G72 P... Q... ;
G73	9	轴向粗车循环	G73 U... R... ;
G74	10	径向粗车循环	G74 W... R... ;
G75	11	轮廓重复	G75 U... W... R... ;
G76	12	轴向切槽多重循环	G76 R... ;
G77	13	径向切槽多重循环	G77 R... ; 或 G77 X(U)... Z(W)... P... Q... R... F... ;
G78	14	多重螺纹切削循环	G78 P... (m, r, a) Q... R... ;
G92	15	设置实际值	G92 (G50) X... Z... ;
G27	16	回参考点运行检查 (准备就绪)	G27 X... Z... ;
G53	17	返回机床坐标系中的位置	(G90) G53 X... Z... ;
G10.6	18	快速提刀打开/关闭	
G5	19	高速循环切削	G05 Pxxxxx Lxxx ;
G30.1	20	参考点位置	
G5.1	21	高速循环 (High-speed cycle) -> 调用 CYCLE305	

## 三种模式的 *G* 代码

### 2.3 SINUMERIK ISO 编程指令车削版模式 C

名称	索引	说明	格式
G92.1	22	删除实际值, 复位 WCS	
G60	23	定向定位	
20. G 代码组 (模态)			
G50.2	1	多边形车削“关”	
G51.2	2	多边形车削“开”	G51.2 P...Q...;
31. G 代码组 (模态)			
G290	1	选择西门子模式	
G291	2	选择 ISO 编程指令模式	

# 运行指令

## 3.1 插补指令

下文为您介绍了定位指令和插补指令，使用这些指令可以根据编写的轮廓（如直线或圆弧）控制刀具轨迹。

### 3.1.1 快速运行(G00)

快速运行可以用于刀具的快速定位、工件的绕行或者移动到换刀位置。

下列 G 功能可以用于定位（参见下表）：

表格 3- 1 定位的 G 功能

G 功能	功能	G 功能组
G00	快速移动	01
G01	直线运行	01
G02	顺时针圆弧/螺线	01
G03	逆时针圆弧/螺线	01

### 使用(G00)定位

#### 格式

G00 X... Y... Z... ;

### 带线性插补的 G00

写入 G00 的刀具运行将以可能的最大速度（快速运行）执行。在每个机床数据中，每个轴的快速运行速度都是单独定义的。如果同时在多个轴上执行快速运行，那么在线性插补时快速运行速度由对轨迹运行所需时间最长的轴来决定。

## 运行指令

### 3.1 插补指令

#### 不带线性插补的 G00

G00 程序段中没有写入的轴也不会运行。定位时每个轴以各自预设的快速移动速度单独运行。机床的精确速度请参见机床制造商的说明资料。

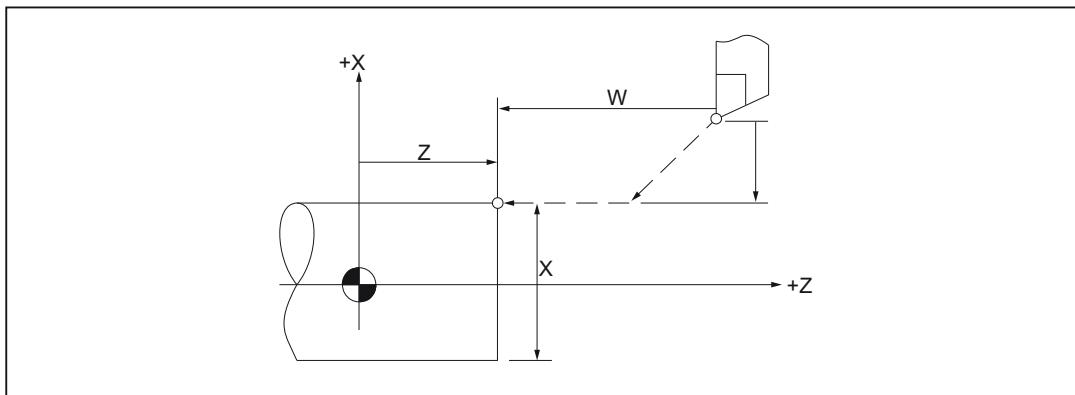


图 3-1 带有 2 个未插补轴的快速运行

#### 说明

由于在 G00 定位时轴单独运行（没有插补），因而每个轴在不同时间到达终点。因此，在多轴定位时要特别仔细谨慎，防止定位时刀具和工件或设备相撞。

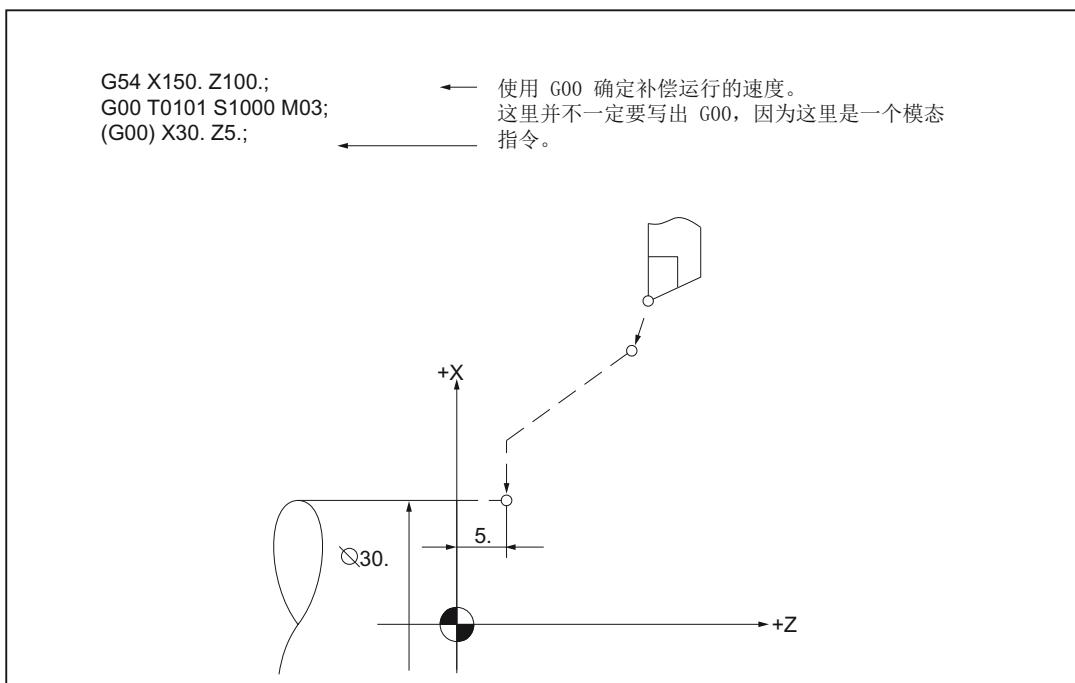


图 3-2 编程示例

## 线性插补(G00)

通过置位机床数据 20732 \$MC\_EXTERN\_GO\_LINEAR\_MODE 可以设置 G00 线性插补。此时，所有编程的轴都以带线性插补的快速运行方式运行，并同时到达目标位置。

### 3.1.2 直线插补 (G01)

使用 G01 刀具以平行于轴的、倾斜的或以空间内任意的直线方式运行。可以用线性插补功能加工 3D 平面、槽等。

#### 格式

G01 X... Z... F... ;

G01 执行带轨迹进给率的线性插补。G01 程序段中没有写入的轴也不会运行。按照上文列举的示例编程线性插补。

#### 轨迹轴进给率 F

进给速度由地址 F 指定。取决于机床数据中的默认设置，G 指令确定的尺寸单位(G93, G98, G99)为毫米或英寸。

允许为每个 NC 程序段编程一个 F 值。通过其中一个 G 指令确定进给速度的单位。进给率 F 只对于轨迹轴有效，并且直到编程新的进给值之前一直有效。地址 F 后允许出现分隔符。

#### 说明

如果在 G01 程序段中或之前的程序段中没有编程任何进给率 Fxx，在执行 G01 程序段时会触发报警。

可以通过绝对值或增量值给定终点。

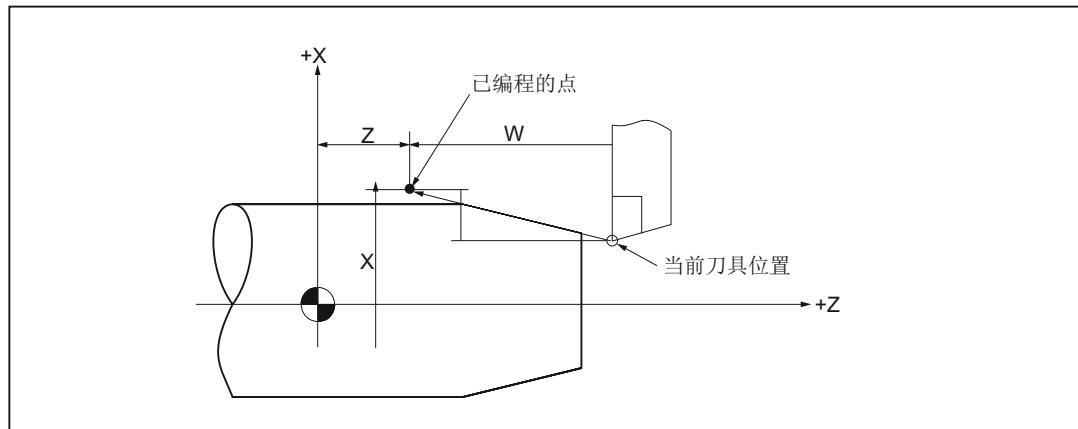


图 3-3 线性插补

## 运行指令

### 3.1 插补指令

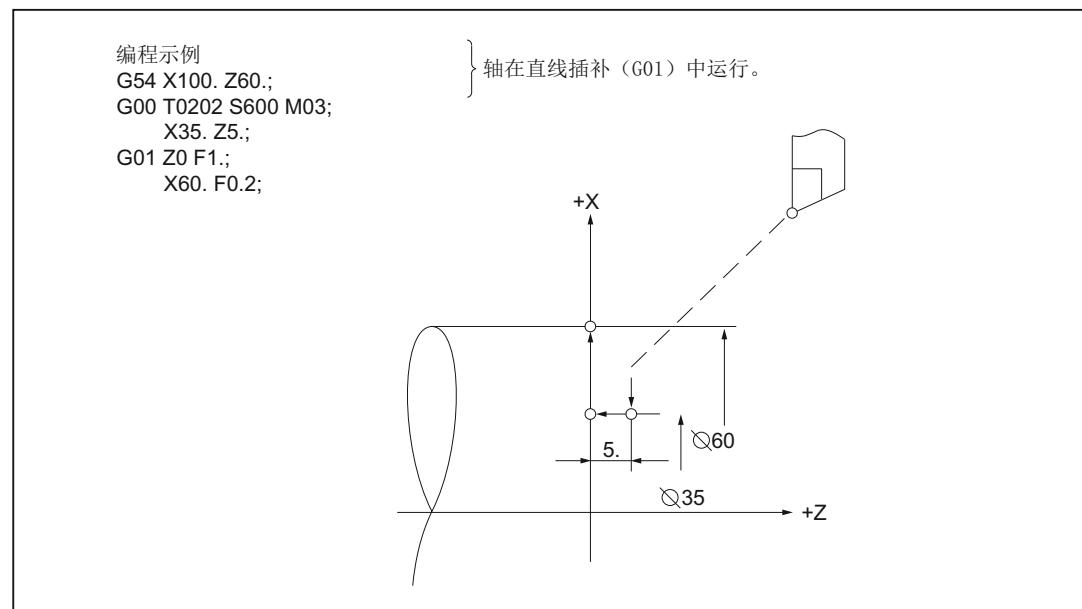


图 3-4 编程示例

### 3.1.3 圆弧插补(G02, G03)

#### 格式

车刀在 ZX 平面中按照以下给定的指令运行所编程的圆弧。所编程的轨迹速度在圆弧上都保持一致。

G02(G03) X(U)... Z(W)... I... K... (R...) F... ;

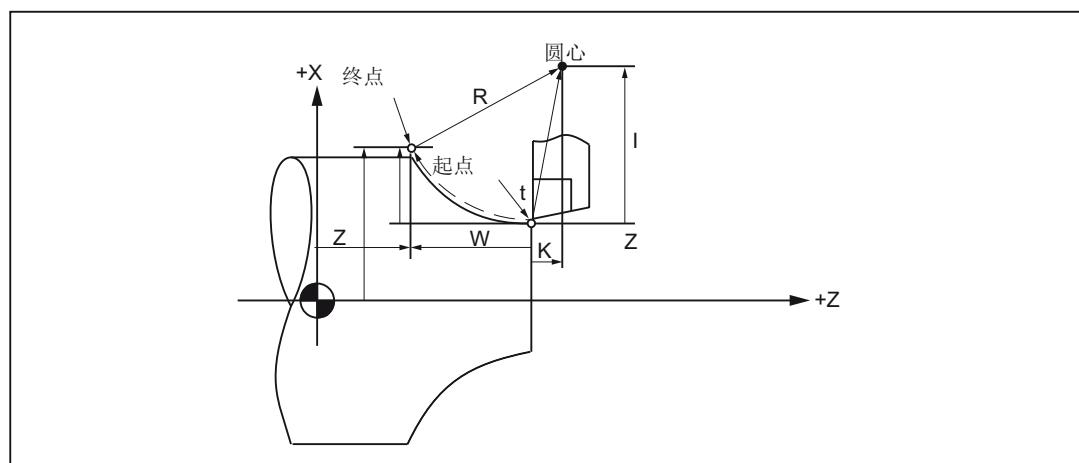


图 3-5 圆弧插补

执行下表列出的指令，启动圆弧插补：

表格 3-2 执行圆弧插补的指令

按键或开关	指令	说明
旋转方向	G02	顺时针方向
	G03	逆时针方向
终点位置	X (U)	圆弧终点的 X 坐标（径向值）
	Z (W)	圆弧终点的 Z 坐标
	Y (V)	圆弧终点的 Y 坐标
起点到中心点的距离	I	在 X 轴方向上圆弧起点与中心点的间距
	J	在 Y 轴方向上圆弧起点与中心点的间距
	K	在 Z 轴方向上圆弧起点与中心点的间距
圆弧的半径	R	起点与圆弧中心点的距离

## 运行指令

### 3.1 插补指令

#### 旋转方向

圆弧的车削方向通过下表中列出的 G 功能来确定。

旋转方向	
G02	顺时针方向
G03	逆时针方向

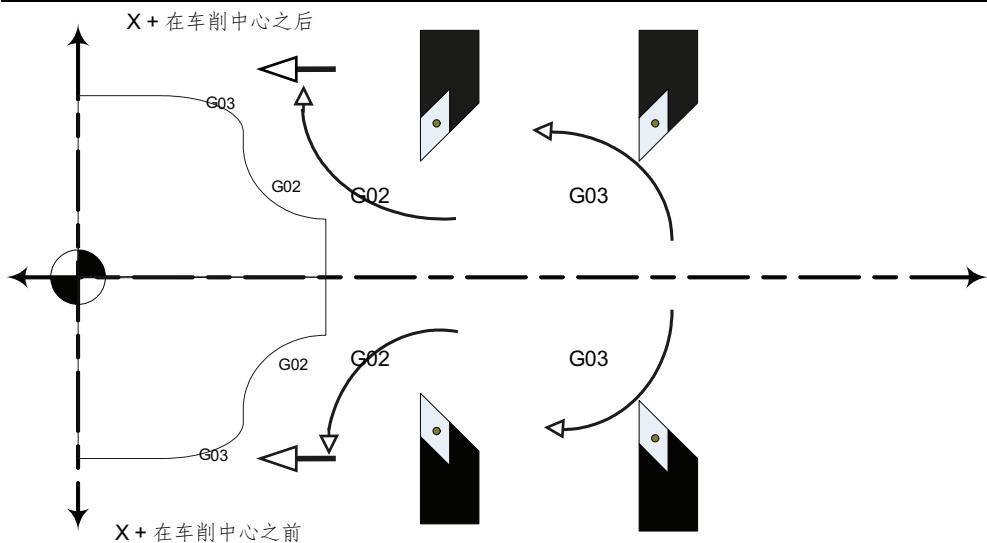


图 3-6 圆弧的旋转方向

## 编程圆弧运行

ISO 模式提供两种编程圆弧运行的方法。

圆弧运行通过以下几点来描述：

- 圆弧中心点和终点，绝对值或增量值
- 以直角坐标表示的半径和终点

对于圆心角  $\leq 180$  度的圆弧插补，应编程“ $R > 0$ ”（正值）。

对于圆心角  $> 180$  度的圆弧插补，应编程“ $R < 0$ ”（负值）。

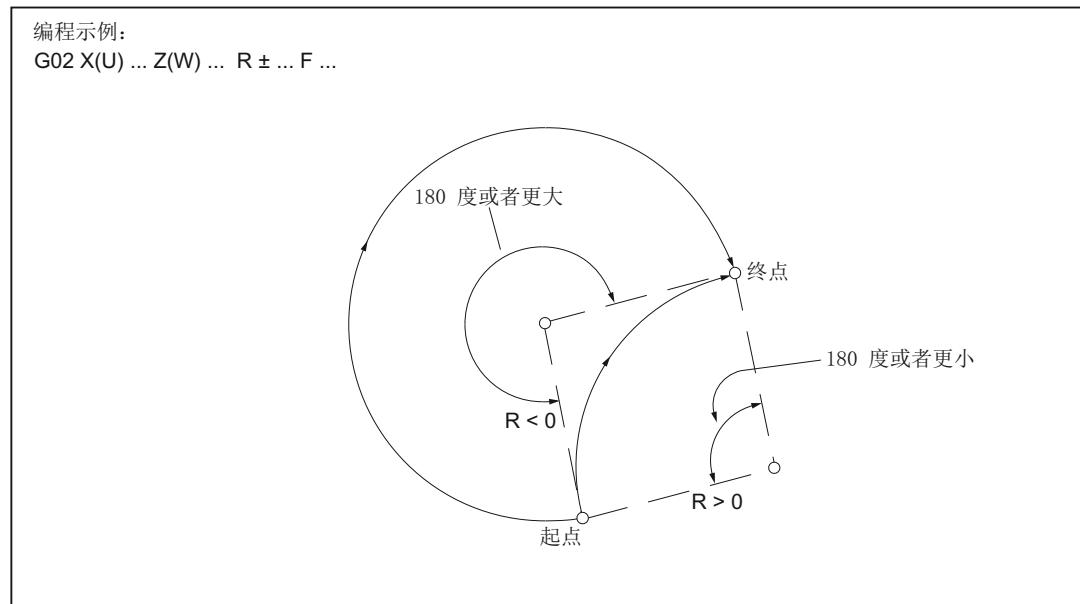


图 3-7 带半径 R 的圆弧插补

## 进给

可以完全按照线性插补中编程的进给率的方式来定义圆弧插补中的进给率（另见章节“线性插补(G01)”）。

## 运行指令

### 3.1 插补指令

#### 编程示例

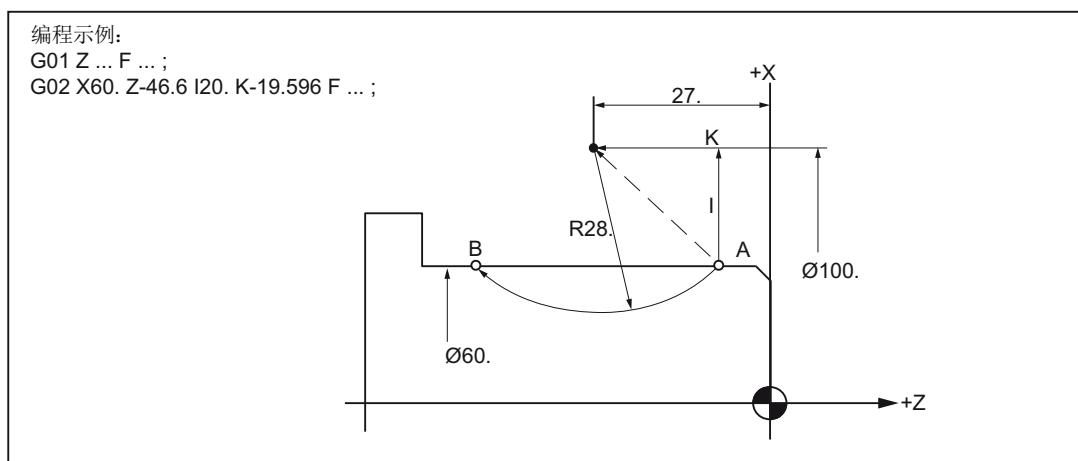


图 3-8 多象限圆弧插补

圆弧中心点	(100.00, 27.00)
“I”值	$\frac{100 - 60}{2} = 20 \text{ mm}$
“K”值	$-\sqrt{28^2 - 20^2} = -\sqrt{384} = -19.596 \text{ mm}$

#### 3.1.4 轮廓段编程以及插入倒角和倒圆

在每个位移程序段后、线性轮廓和圆弧轮廓之间可以插入倒角或倒圆，例如：用于倒去工件边缘锋利的毛刺。

可以在以下轮廓组合中插入倒角或倒圆：

- 两条直线之间
- 两段圆弧之间
- 一段圆弧和一条直线之间
- 一条直线和一段圆弧之间

#### 格式

, C...; 倒角

, R...; 倒圆

## 示例

```
N10 G1 X9. Z100. F1000 G18
G1 X19 Z100
X31 ANG=140 CHF=7.5
N30 X80. Z70., A95.824, R10
```

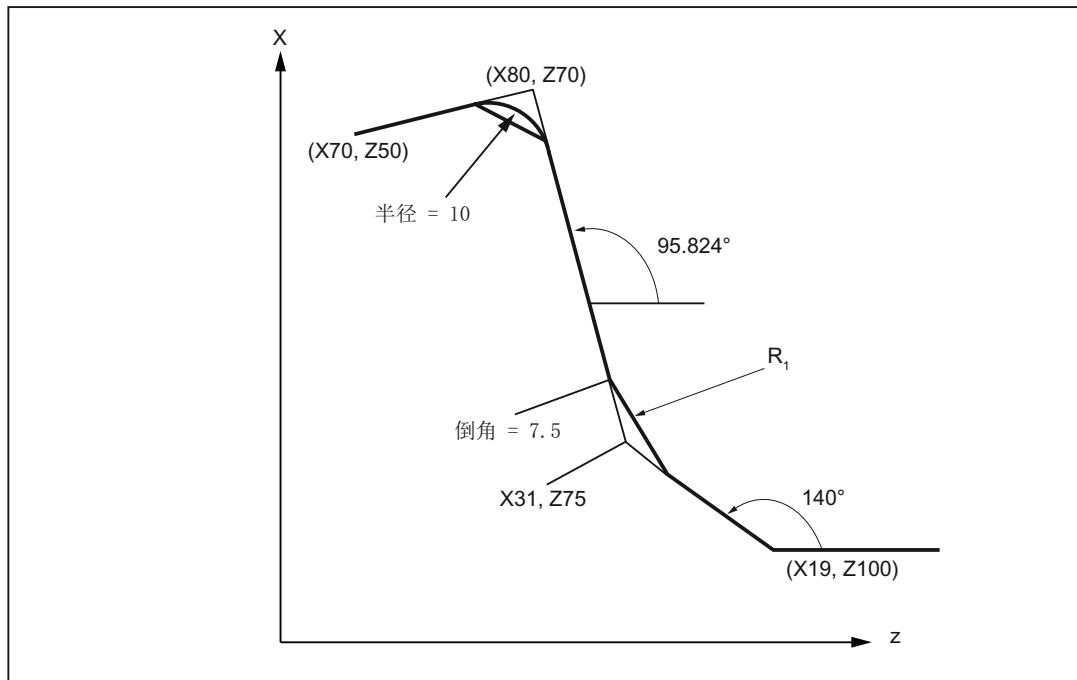


图 3-9 3 条直线

## ISO 编程指令模式

在 ISO 原始语言中，地址 C 不仅可以用作轴名称，也可以用作轮廓倒角的名称。

地址 R 不仅可以是一个循环参数，也可能是轮廓半径的标识符。

为加以区分，写入轮廓段时必须在地址“R”或“C”前加上逗号“，”。

## 运行指令

### 3.2 使用 G 功能回参考点

## 3.2 使用 G 功能回参考点

### 3.2.1 通过中间点回参考点(G28)

#### 格式

G28 X... Z... ;

通过指令“G28 X(U)...Z(W)...C(H)...Y(V);”可以使编程的轴回到参考点。其中，编程的轴首先以快速运行方式运行到给定位置，然后从该位置自动运行到参考点。G28 程序段中未编程的轴不会运行到其参考点。

#### 参考点位置

在机床开机后，如果使用的是增量式位移测量系统，则所有轴必须回到参考点标记。在此之后，才可以编程运行位移。用 G74 可以在 NC 程序中执行回参考点运行。参考点坐标由机床数据 34100 \$\_MA\_REFP\_SET\_POS[0] 到 [3] 来定义。一共可以确定四个参考点位置。

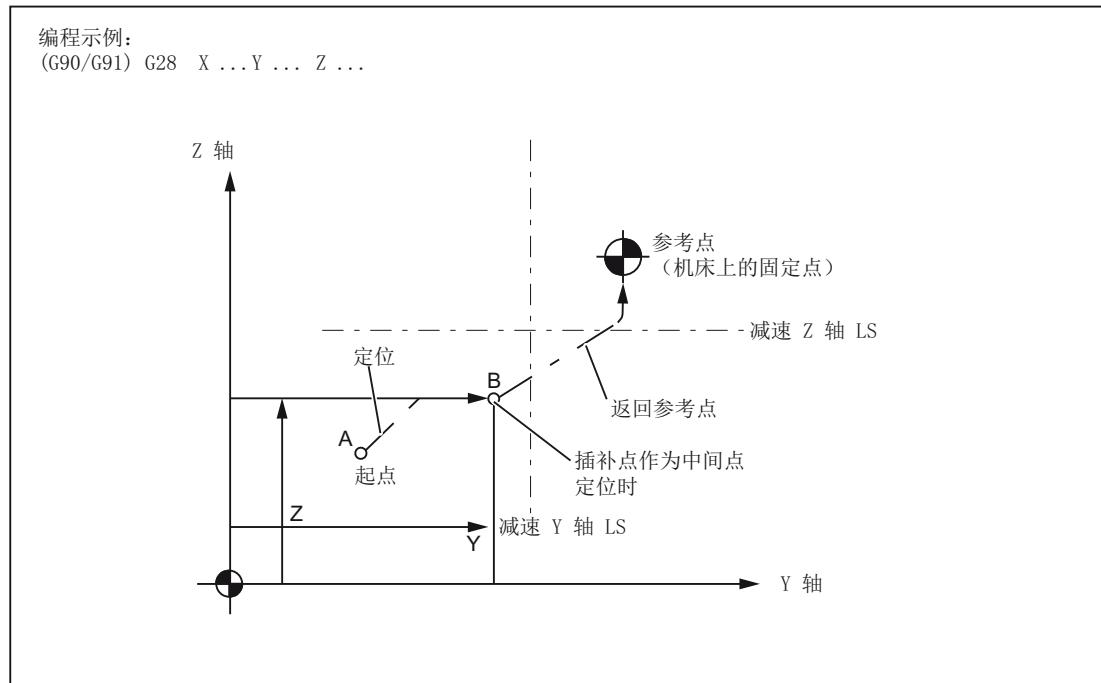


图 3-10 自动回参考点

#### 说明

功能 G28 由 shell 循环 cycle328.spf 实现。

如果需要通过 G28 使轴运行到参考标记，在回参考点运行之前不允许编写该轴的转换。通过 cycle328.spf 的指令 TRAFOOF 取消转换。

### 3.2.2 参考位置点检查(G27)

#### 格式

G27 X... Z... ;

通过该功能可以检查轴是否在参考点上。

#### 检查过程

如果通过 G27 成功执行检查，则继续处理下一个零件程序段。如果一个 G27 编写的轴没有位于参考点上，则输出报警 61816“轴不在参考点上”，并中断自动运行。

#### 说明

和 G28 一样，功能 G27 由 shell 循环 cycle328.spf 实现。

在执行 G27 前应该取消选择功能“镜像”，以避免定位错误。

### 3.2.3 通过参考点选择回参考点(G30)

#### 格式

G30 Pn X... Z... ;

在执行指令“G30 Pn X... Y... Z;”时轴以连续路径加工定位到给定的中间点，然后运行到 P2 - P4 选中的参考点。“G30 P3 X30.;”时 X 轴和 Y 轴返回第三参考点。如果省略了“P”，则第二参考点被选中。G30 程序段中没有写入的轴也不会运行。

#### 参考点位置

所有参考点的位置始终以第一个参考点为基准。第一个参考点和其他所有参考点的间距显示在下列机床数据中：

表格 3-3 参考点

按键或开关	MD
2. 参考点	\$_MA_REFP_SET_POS[1]
3. 参考点	\$_MA_REFP_SET_POS[2]
4. 参考点	\$_MA_REFP_SET_POS[3]

#### 说明

写入 G30 时需注意的细节请参见章节“通过中间点回参考点(G28) (页 36)”。和 G28 一样，功能 G30 由 shell 循环 cycle328.spf 实现。

## 运行指令

### 3.3 切削螺纹功能的应用

## 3.3 切削螺纹功能的应用

### 3.3.1 等螺距螺纹切削 (G32)

#### 格式

通过指令“G32 X (U)... Z (W)... F... ;”可将“圆柱螺纹”，“平面螺纹”，“圆锥螺纹”这三种螺纹类型作为右螺纹或左螺纹进行加工。通过 F 确定螺距。终点坐标通过 X, Z (绝对) 或 U, W(增量)确定。

G 代码系统 A	G 代码系统 B	G 代码系统 C
G32	G33	G33

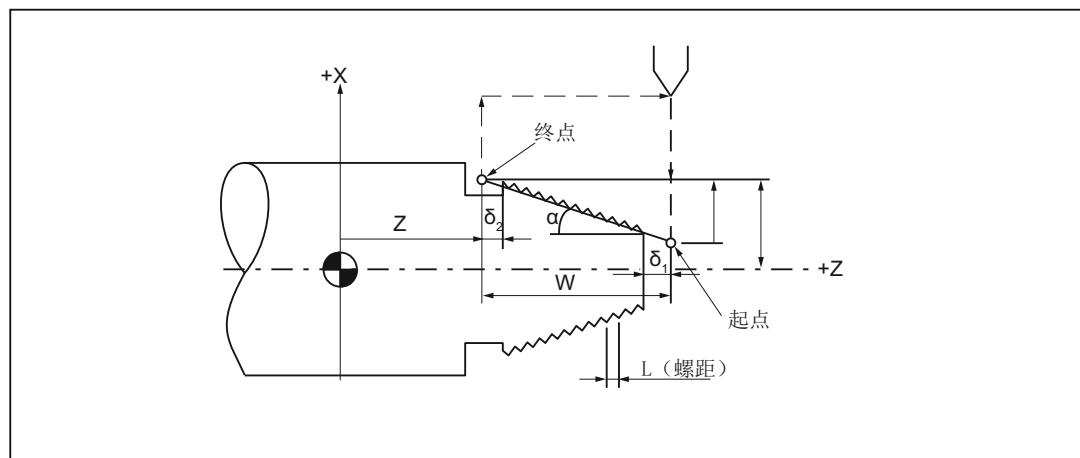


图 3-11 螺纹切削

#### 螺距方向

在圆锥螺纹上，圆锥角决定了所编程螺距的生效方向。

表格 3- 4 螺距方向

		螺距方向
	$\alpha \leq 45^\circ$	编程的螺距在 Z 轴方向上生效。
	$\alpha > 45^\circ$	编程的螺距在 X 轴方向上生效。

## 示例

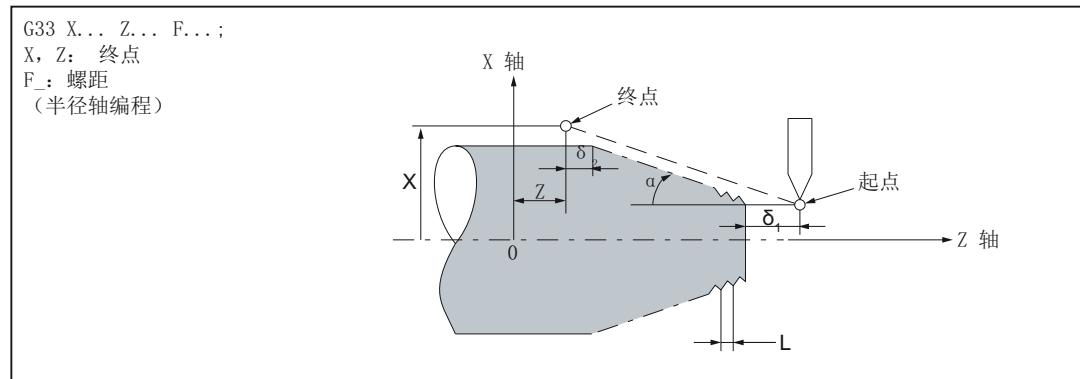


图 3-12 编程示例

## 圆柱螺纹切削的示例 (G 代码系统 A)

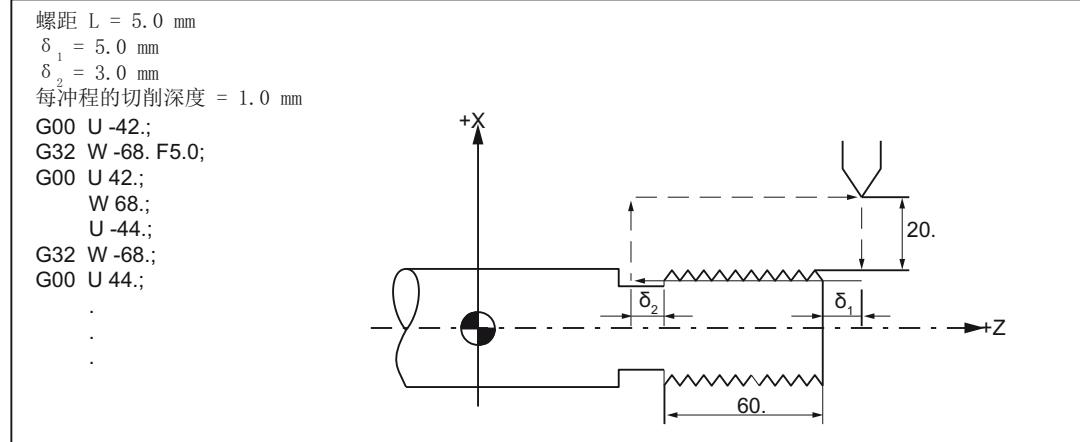


图 3-13 圆柱螺纹切削的编程示例

## 圆锥螺纹切削的示例 (G 代码系统 A)

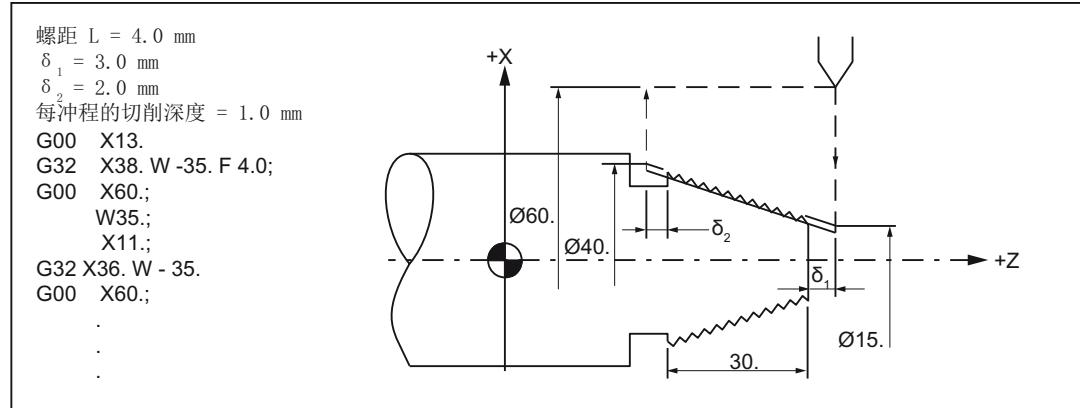


图 3-14 圆锥螺纹切削的编程示例

### 3.3 切削螺纹功能的应用

#### 前提条件:

技术上的前提条件是，一个带有行程测量系统并可以进行转速调节的主轴

#### 操作步骤:

数控系统根据编程的主轴转速和螺纹螺距计算出必要的进给率，车刀按此进给率纵向和/或横向穿过螺纹长度。进给率 F 在 G32 时不用考虑，对于最大轴速度（快速运行）的限制由数控系统进行监控。

### 3.3.2 螺纹链接 (G32)

通过连续编程多个 G32 程序段，可依次将多个螺纹组排列成链。通过用 G64 连续路径模式，以预读速度控制连接各程序段，从而避免产生速度突变。

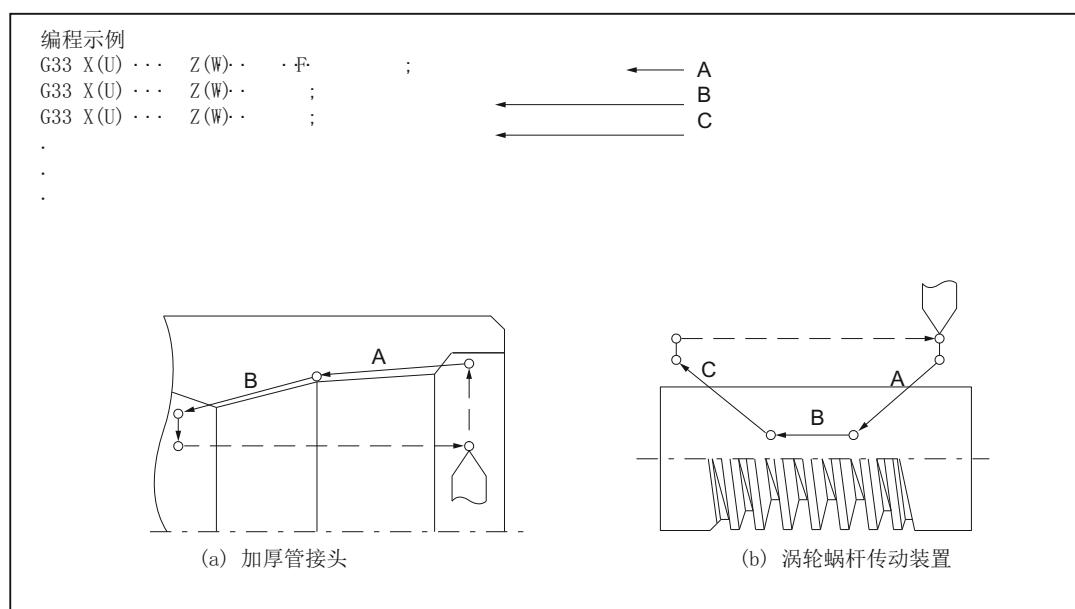


图 3-15 连续螺纹切削

#### 说明

螺纹切削还未结束时，不能改变主轴转速！如果主轴转速不能保持恒定，就会存在由于出现同步误差而导致精度降低的风险。

#### 说明

在螺纹切削时不考虑进给影响和进给停止！

如果在 G98 (每分钟进给) 运行时写入了指令 G32，则触发报警。

### 3.3.3

### 多头螺纹切削 (G32)

通过确定起点偏移对多头螺纹进行加工。起点偏移在地址 Q 上被定义为绝对角度位置。

对其所属的设定数据 42000 (\$SD\_THREAD\_START\_ANGLE) 进行相应的更改。

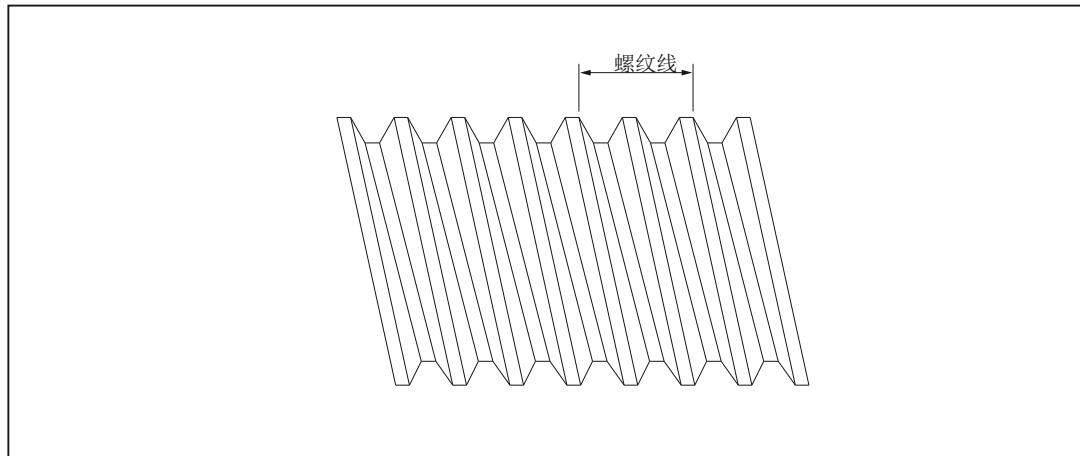


图 3-16 双头螺纹

#### 格式

通过指令“G32 X (U)... Z (W)... F... Q... ;”起点脉冲输出后主轴按照由地址符 Q 给定的角度旋转。接着以 X (U) 和 Z (W) 给定的终点的方向并以地址符 F 给定的螺距开始进行螺纹切削。

多头螺纹切削时地址 Q 的设定:

最小输入增量: 0.001°

可编程范围:  $0 \leq B < 360.000$

## 运行指令

### 3.3 切削螺纹功能的应用

#### 多头螺纹切削时起始角的计算

一般通过设定数据 \$SD\_THREAD\_START\_ANGLE 确定螺纹切削的起始点。多头螺纹切削时，通过按螺线数量划分  $360^\circ$  来计算各个起始点之间的角度偏移。下图为多头螺纹（双头，三头，四头）的示例。

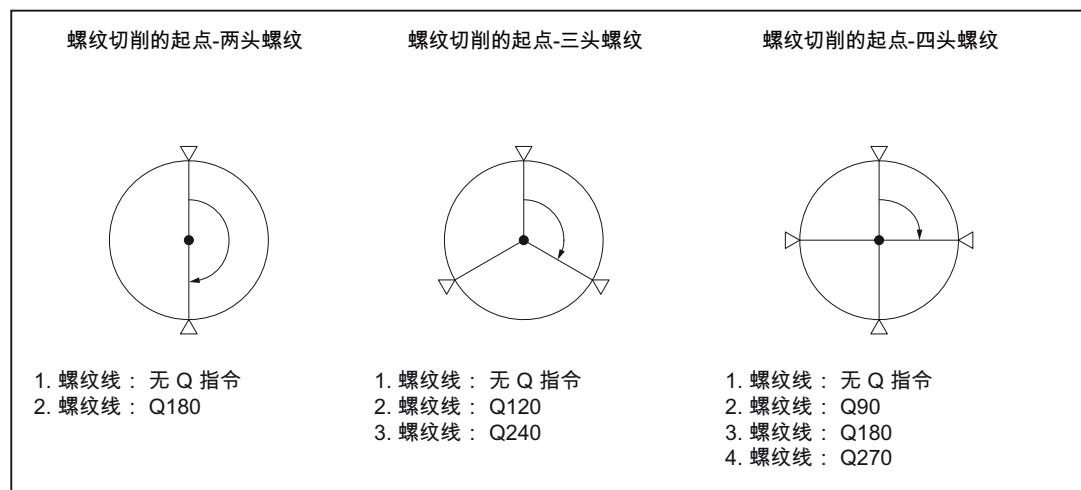


图 3-17 多头螺纹切削时起始角的计算

#### 多头螺纹的编程示例（G 代码系统 A）

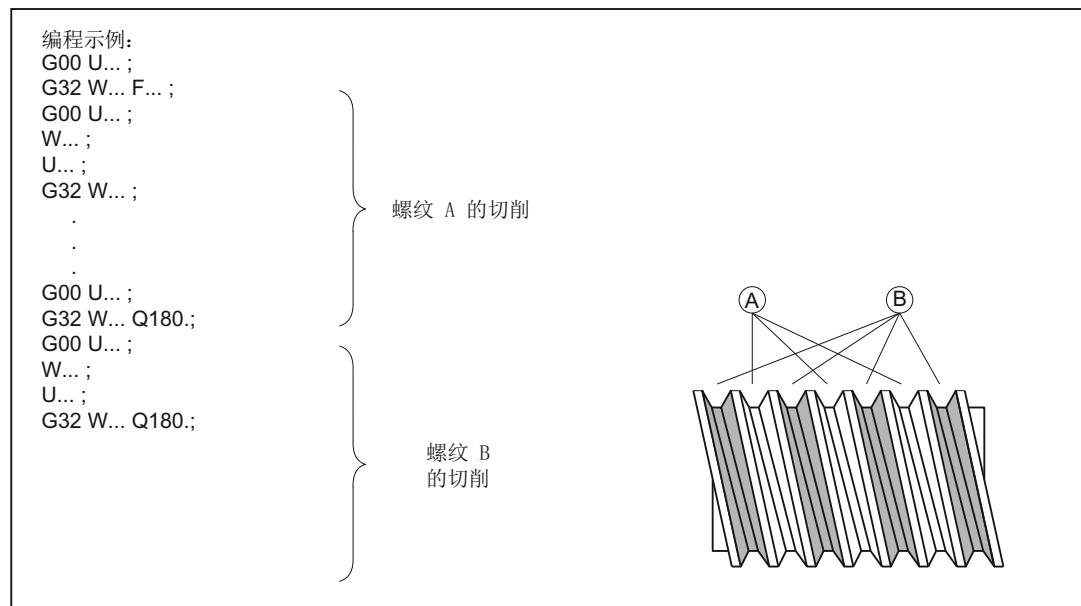


图 3-18 主轴旋转角的给定

**说明**

如果没有（通过 Q）给出起始点偏移，那么使用在设置数据中确定的“螺纹起始角”。

### 3.3.4 可变螺距螺纹切削（G34）

通过指令“G34 X (U)... Z (W)... F... K... ;”进行可变螺距的螺纹切削；通过地址 K 设定每个主轴旋转的螺距变化。

**格式**

G34 X... Z... F... K... ;

G 代码系统 A	G 代码系统 B	G 代码系统 C
G34	G34	G34

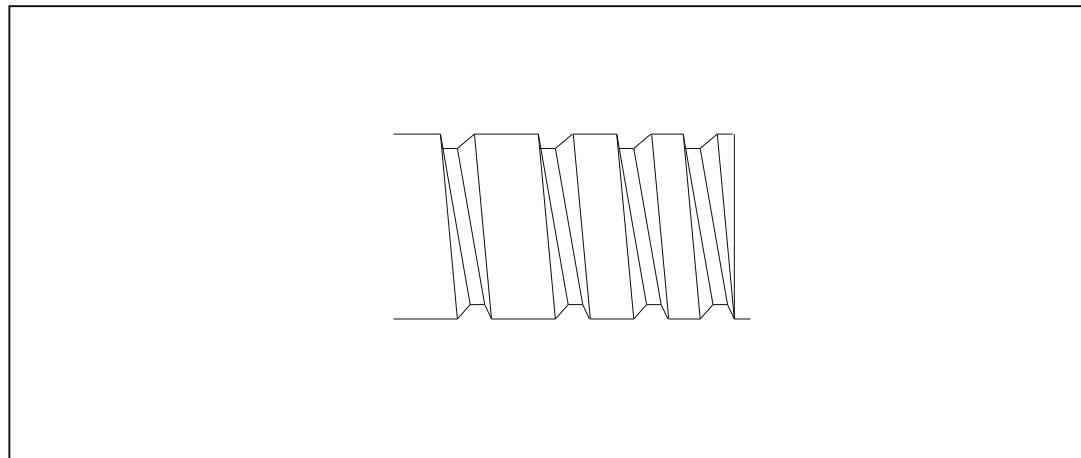


图 3-19 可变螺距螺纹

**终点处进给速度**

设定指令，确保终点处的进给率不为负值！

$$(F + \frac{K}{2})^2 + 2KW > 0$$

### 3.3 切削螺纹功能的应用

#### 计算螺距变化

如果已知螺纹的起始螺距和最终螺距，那么就可以根据下面的等式计算出要编程的螺距变化：

$$F = \frac{|k2e - k2a|}{2 * IG[\text{mm/U}]}$$

这些标识符具有下列含义：

**K2e:** 以[毫米/转]表示的轴目标点坐标的螺纹距

**K2a:** 以[毫米/转]表示的螺纹起始螺距(在 I, J, K 下编程)

**IG:** 螺纹长度，单位 [mm]

# 测量指令

## 4.1

### 坐标系

刀具的位置由它在坐标系中的坐标明确定义。该坐标由轴的位置定义。例如如果两个参与运行的轴为 X 轴和 Z 轴，则以如下方式给出坐标：

X... Z...

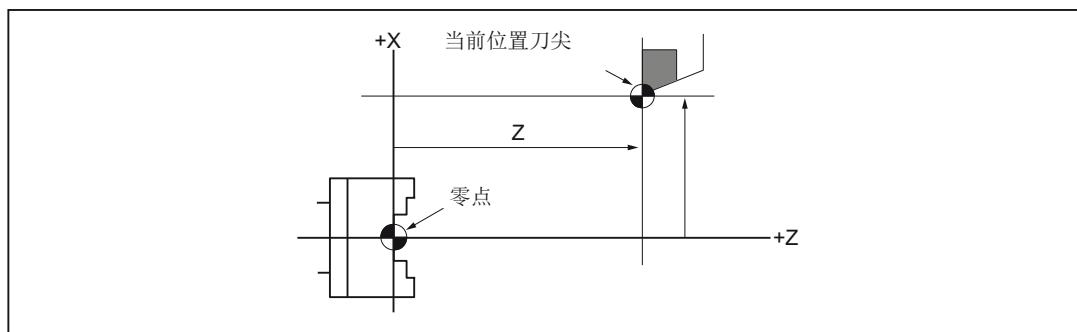


图 4-1 通过 X... Z..给定的刀具位置

可以使用以下坐标系设定坐标：

1. 机床坐标系(G53)
2. 工件坐标系(G50)
3. 局部坐标系(G52)

## 4.1 坐标系

### 4.1.1 机床坐标系(G53)

#### 确定机床坐标系

使用机床零点可以确定机床坐标系 MCS。所有其他参考点都以机床零点为基准。

机床零点是机床上的固定点，所有测量系统都以此点为出发点。

#### 格式

G53 X... Z... ;

X, Z: 绝对值指令

#### 选择机床坐标系(G53)

G53 以程序段方式抑制可编程和可设定的零点偏移。如果需要使刀具运行到某个机床特定位置，则始终以 G53 为基础在机床坐标系中对运行进行编程。

#### 取消补偿

如果 MD10760 \$MN\_G53\_TOOLCORR = 0, G53 程序段中有效的刀具长度补偿和刀具半径补偿保持生效

如果 \$MN\_G53\_TOOLCORR = 1, G53 程序段中有效的刀具长度补偿和刀具半径补偿被抑制。

### 4.1.2 工件坐标系(G50)

在开始加工前应为工件创建坐标系，即工件坐标系。在本段落中将为您介绍各种设置、取消和修改工件坐标系的方法。

#### 设置工件坐标系

您可以通过以下两种方法设置工件坐标系：

1. 通过 G50 (G 代码系统 B 和 C 中的 G92) 设置
2. 通过 HMI 操作界面手动选择

#### 格式

G50 (G92) X... Z... ;

#### 说明

通过 G50 对从基本坐标系 (BCS) 向基本零点系统 (BNS) 的坐标转换进行编程。G50 的功能等同可设定的零点偏移。

### 4.1.3 复位刀具坐标系(G50.3)

通过 G50.3 X.. (G 代码系统 B 和 C 为 G92.1 P0) 可以将已经偏移的坐标系复位到偏移前的状态。从而可以使工件坐标系恢复为有效可设定零点偏移(G54-G59)定义的坐标系。如果没有有效的可设定零点偏移，则工件坐标系被设为参考位置。G50.3 使由 G50 或 G52 执行的偏移复位。但只有编写该功能的轴才被复位。

示例 1:

N10 G0 X100 Y100	; 显示: WCS: X100 Y100	MCS: X100 Y100
N20 G50 X10 Y10	; 显示: WCS: X10 Y10	MCS: X100 Y100
N30 G0 X50 Y50	; 显示: WCS: X50 Y50	MCS: X140 Y140
N40 G50.3 X0 Y0	; 显示: WCS: X140 Y140	MCS: X140 Y140

示例 2:

N10 G10 L2 P1 X10 Y10		
N20 G0 X100 Y100	; 显示: WCS: X100 Y100	MCS: X100 Y100
N30 G54 X100 Y100	; 显示: WCS: X100 Y100	MCS: X110 Y110
N40 G50 X50 Y50	; 显示: WCS: X50 Y50	MCS: X110 Y110
N50 G0 X100 Y100	; 显示: WCS: X100 Y100	MCS: X160 Y160
N60 G50.3 X0 Y0	; 显示: WCS: X150 Y150	MCS: X160 Y160

### 4.1.4 选择工件坐标系

如上所述，用户可以从已经设置的工件坐标系中选出一个坐标系。

#### 1. G50

只有当此前选择了一个工件坐标系时，工件坐标系中的绝对值指令才生效。

#### 2. 通过 HMI 操作界面从给定的工件坐标系中选择一个工件坐标系

输入 G54 到 G59 范围内的 G 功能可以选择一个工件坐标系。

开机回参考点后工件坐标系建立。在 MD20154[13]中设置坐标系的闭合位置。

#### 4.1.5 写入零点偏移/刀具补偿(G10)

可以通过两种方法修改由 G54 到 G59 或 G54 P{1 ... 93} 定义的工件坐标系。

1. 通过 HMI 操作面板输入数据
2. 通过程序指令 G10 或 G50 (设置实际值)

#### 格式

通过 G10 修改:

G10 L2 Pp X (U)... Y(V)... Z(W)... ;

p=0: 外部工件零点偏移

p=1 到 6: 工件零点偏移值和工件坐标系 G54 到 G59 对应 (1 = G54 到 6 = G59)

X, Y, Z: 工件坐标系偏移的绝对设置数据。

U, V, W: 工件坐标系偏移的增量设置数据。

G10 L20 Pp X (U)... Y(V)... Z(W)... ;

p=1 到 93: 工件零点偏移值和工件坐标系 G54 P1 ... P93 对应。可以通过 MD18601

\$MN\_MM\_NUM\_GLOBAL\_USER\_FRAMES 或 MD28080

\$MC\_MM\_NUM\_USER\_FRAMES 设定零点偏移的数量 (1 到 93)。

X, Y, Z: 工件坐标系偏移的绝对设置数据。

U, V, W: 工件坐标系偏移的增量设置数据。

通过 G50 修改:

G50 X... Z... ;

## 说明

通过 G10 修改:

通过 G10 可以单独修改每个工件坐标系。如果首次通过 G10 写入零点偏移，在机床上执行 G10 程序段（主运行程序段）时，必须置位 MD20734

\$MC\_EXTERN\_FUNCTION\_MASK, 位 13。然后通过 G10 执行内部 STOPRE。通过该机床数据位可以控制 ISO 编程指令 T 和 ISO 编程指令 M 的所有 G10 指令。

通过 G50 修改:

通过给定 G50 X... Z... 可以移动之前由 G 指令（G54 到 G59 或 G54 P{1 ...93}）选择的工件坐标系，并可设置一个新的工件坐标系。如果以增量值写入 X、Y 和 Z，则必须定义恰当的坐标系；其中，当前刀具位置和给定增量值之和应该等于前一刀具位置的坐标值（坐标系偏移）。接着坐标系的偏移值和每个工件零点偏移值相加。也就是说：整个工件坐标系按照相同的值移动。

## 示例

刀具以 G54 定位到(190, 150)，每次 G50X90Y90 时工件坐标系 1 (X' - Y')产生偏移矢量 A。

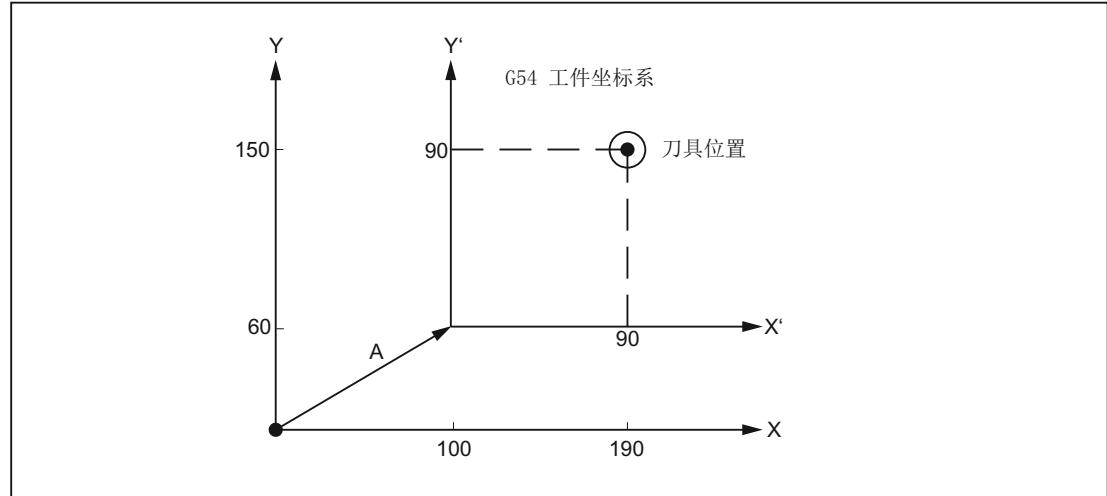


图 4-2      示例：设置坐标

## 4.2 确定坐标值的输入方式

### 4.2.1 X 轴的直径和半径编程

使用地址 X 或 U 进行 X 轴的指令编程:

如果通过机床数据 20110 \$MC\_DIAMETER\_AX\_DEF = "X" 将 X 轴定义为横向轴, 并且通过 MD20150 \$MC\_GCODE\_RESET\_VALUES[28] = 2 激活直径编程 (= 西门子 G 代码 DIAMON), 则所编程的轴位置被视为直径值。

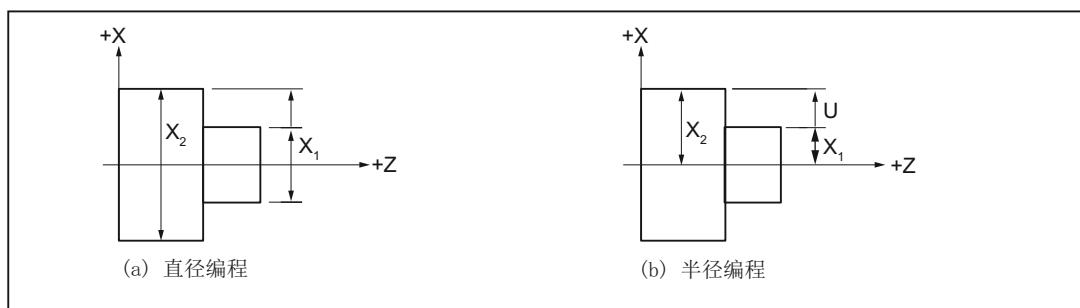


图 4-3 坐标值

直径值对于下列数据有效:

- 工件坐标系中端面轴的实际值显示
- JOG-运行: 增量尺寸的增量和手轮运行
- 从结束位置进行编程

## 4.2.2 英制/公制尺寸输入(G20, G21)

取决于生产图纸的尺寸标注方式，编程时可以为和工件相关的几何轴选择公制或英制尺寸。通过以下 G 功能可以选择输入单位：

表格 4-1 用于选择尺寸单位的 G 功能

G 功能	功能	G 功能
G20 (G70, G 代码系统 C)	“英寸”输入	06
G21 (G71, G 代码系统 C)	“毫米”输入	06

### 格式

必须始终在程序段开头处写入 G20 和 G21，并且该程序段中不允许出现其它指令。

### 关于英制/公制转换的补充说明

下面的几何参数可以由数控系统（带必要的偏差）换算为没有设定的尺寸系统，并可以直接输入：

#### 举例

- 路径信息 X, Z
- 插补参数 I, J, K 和圆弧编程的圆弧半径 R
- 螺距 (G32, G34)
- 可编程的零点偏移

#### 说明

所有其它的参数，诸如进给、刀具补偿或者可设定零点偏移（使用 G20/G21 时）以尺寸系统的基本设定 (MD10240 SCALING\_SYSTEM\_IS\_METRIC) 说明。

系统变量和机床数据的描述同样与 G20/G21 程序状态无关。如果要求 G20/G21 中的进给值生效，则必须清楚地编程一个新的 F 值。

### 参考文献：

/FB1/功能手册 基本功能：速度、额定/实际值系统、闭环控制(G2)，章节“公制/英制单位系统”

表格 4-2 G20 或 G21 运行时的刀具补偿量

保存的刀具补偿量	G20 运行 (单位“英寸”)	G21 运行 (单位“毫米”)
150000	1.5000 英寸	15.000 mm

## 4.3 控制时间的指令

通过 G04 可以在两个 NC 程序段间写入工件加工的中断时间或主轴转数，从而可以进行自由切削。

通过 MD20734 \$MC\_EXTERN\_FUNCTION\_MASK，位 2 可以设定暂停时间以时间（秒或毫秒）为单位还是以主轴转数为单位。如果设置了 \$MC\_EXTERN\_FUNCTION\_MASK，位 2=1，G98 生效时暂停时间以秒为单位；如果选择了 G99，暂停时间以主轴转数[转]为单位。

### 格式

G04 X...; 或 G04 P...;

X\_：时间值（可以为小数）

P\_：时间值（不能为小数）

- 必须在一个单独的程序段中写入暂停时间(G04 ..)。

有两种方法可用于执行写入的暂停时间：

MD \$MC\_EXTERN\_FUNCTION\_MASK

位 2 = 0: 暂停时间总以单位“秒”[s]设定

位 2 = 1: 以单位“秒”(G98 有效) 或以主轴转数 (G99 有效) 设定暂停时间

通过设定 G98 (每分钟进给率)，下一个程序段的加工会相应地延迟一定时间（单位“秒”），而设定 G99 (旋转进给率) 时则会延迟一定的主轴旋转次数。

必须在程序段中单独写入 G04。

### 示例

G98 G04 X1000 ;

标准记数法：  $1000 * 0.001 = 1$  秒

计算器记数法： 1000 秒

G99 G04 X1000 ;

标准记数法：  $1000 * 0.001 = 1$  主轴转数

计算器记数法： 1000 主轴转数

## 4.4

## 刀具补偿功能

将每个刀具的刀具数据分别输入数控系统的刀具补偿数据存储器中。

您可以直接编程工件尺寸，比如加工图纸中如何标注就可以如何编程。

在加工工件时会自动考虑刀具的几何数据，使其使用各自的刀具就能够加工出编程的轮廓。

### 4.4.1

### 刀具补偿数据存储器

将每个刀具的刀具数据分别输入数控系统的刀具补偿数据存储器中。在程序中只调用所需的刀具及其刀具补偿数据。

## 目录

几何尺寸：长度，半径

尺寸包含多个分量（几何量，磨损量）。数控系统从这些部分再计算出最后的尺寸（比如总长度 1，总半径）。补偿存储器激活时，相应的总尺寸生效。

在进给轴中如何计算这些值，由刀具类型和平面选择指令 G17, G18, G19 决定。

## 刀具类型

刀具类型决定需要哪些几何参数，以及如何进行计算（钻头，或者车刀，或者铣刀）。

## 刀沿长度

对于车刀，还需另外说明刀沿长度。下图给出了所需的刀具参数的信息。

#### 4.4.2 刀具长度补偿

该值补偿所使用刀具长度之间的差异。

刀具的长度是指刀架基准点与刀尖之间的距离。

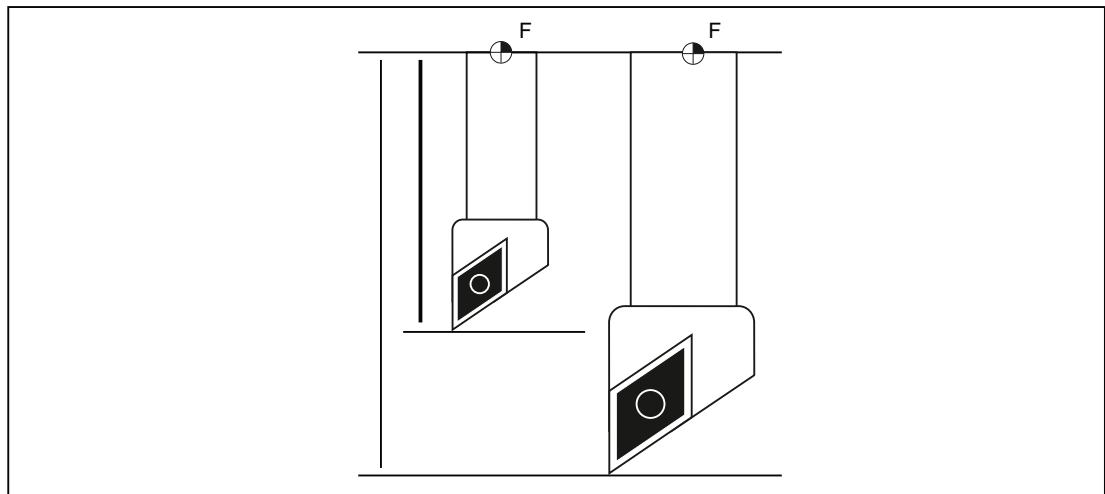


图 4-4 刀具长度

测量出这个长度，然后与可设定的磨损量一起输入到刀具补偿数据存储器中。数控系统就据此计算出进刀时的移动量大小。

#### 4.4.3 刀沿半径补偿 (G40, G41/G42)

因为刀具刀尖一般为圆形，因此若不考虑刀沿半径，在圆锥车削或圆弧加工时会产生轮廓不精确的问题。存在这些问题时，显示下图。通过 G41 或 G42 激活刀沿半径补偿，用于消除轮廓的不精确。

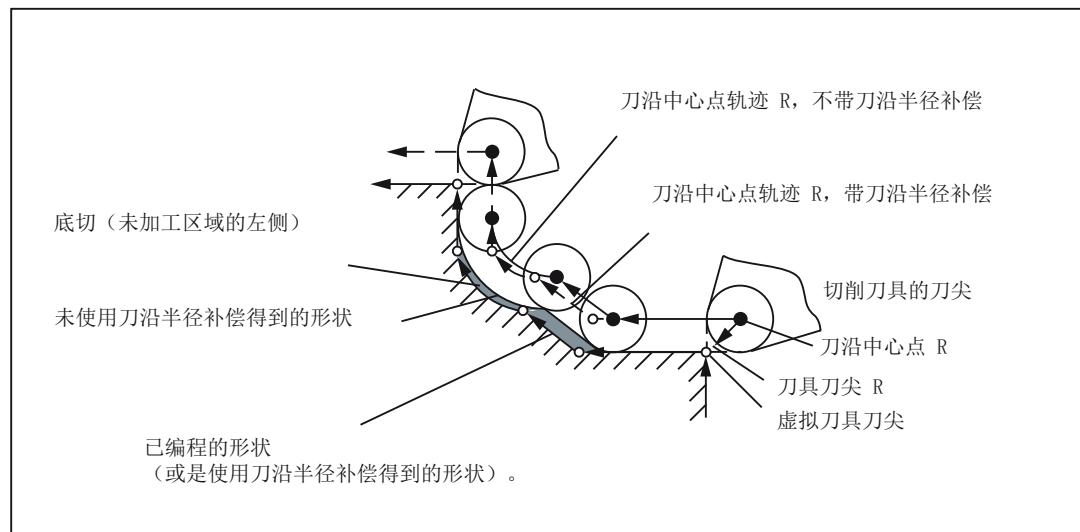


图 4-5 无刀沿半径补偿的加工

#### 刀沿半径补偿量

术语“刀沿半径补偿量”是指刀具刀尖和刀沿中心点 R 之间的距离。

- 确定刀沿半径补偿量

通过不带符号的刀尖圆弧半径设定刀沿半径补偿量。

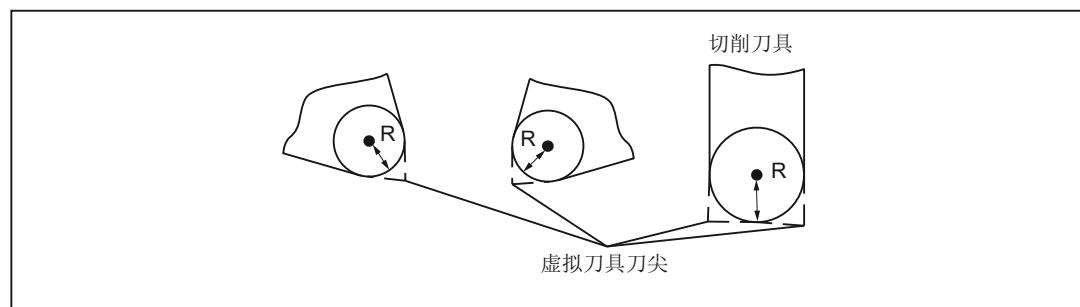


图 4-6 确定刀沿半径补偿量和虚拟刀具刀尖

## 测量指令

### 4.4 刀具补偿功能

#### 确定虚拟刀具刀尖位置（控制点）

- 控制点存储器

从刀具刀尖的中心点 R 处进行观察，并以一个 0 至 9 之间的一位数设定虚拟刀具刀尖的位置。该点为控制点。存储刀具数据前应将控制点输入数控系统存储器。

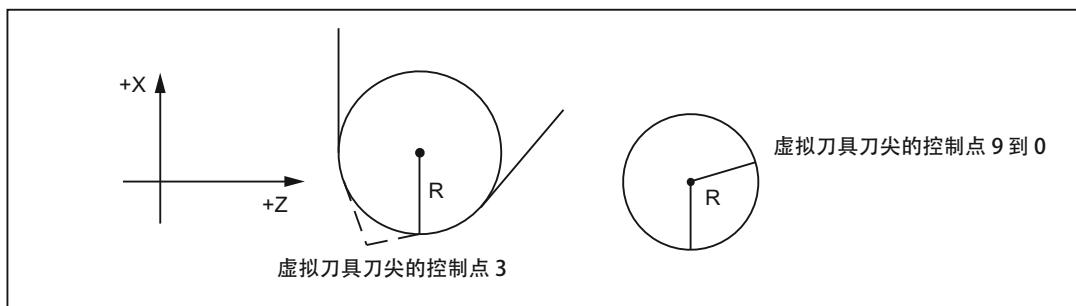


图 4-7 确定控制点的示例

#### 控制点和程序

通过使用控制点 1 至 8，刀具刀尖的虚拟位置必须在编写程序时用作参考点。只有在确定坐标系后才能编写程序。

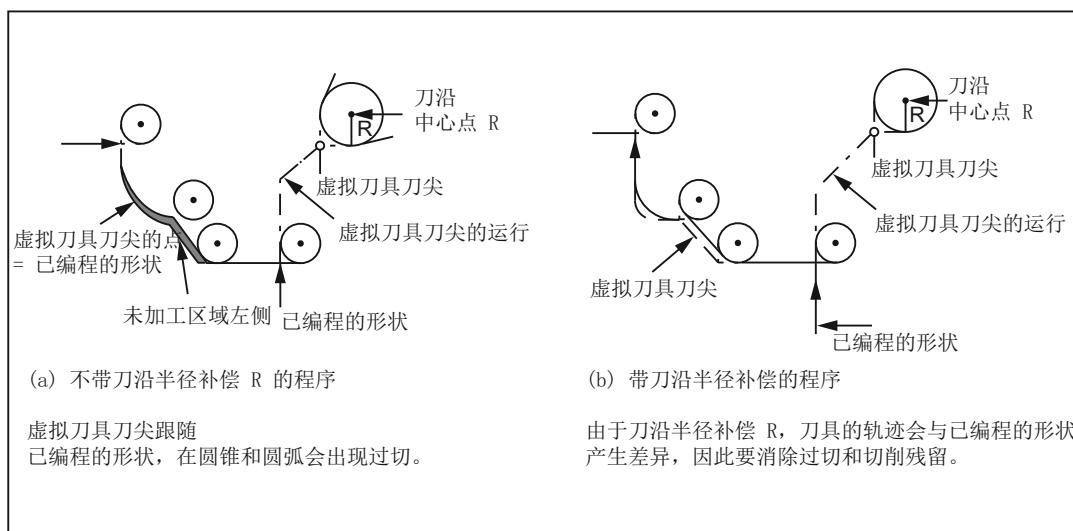


图 4-8 控制点 1 至 8 的程序和刀具运行

使用控制点 0 至 9 时刀沿中心点 R 必须在编写程序时用作参考点。只有在确定坐标系后才能编写程序。如果未使用刀沿半径补偿，则编程形状不能和加工形状不同。

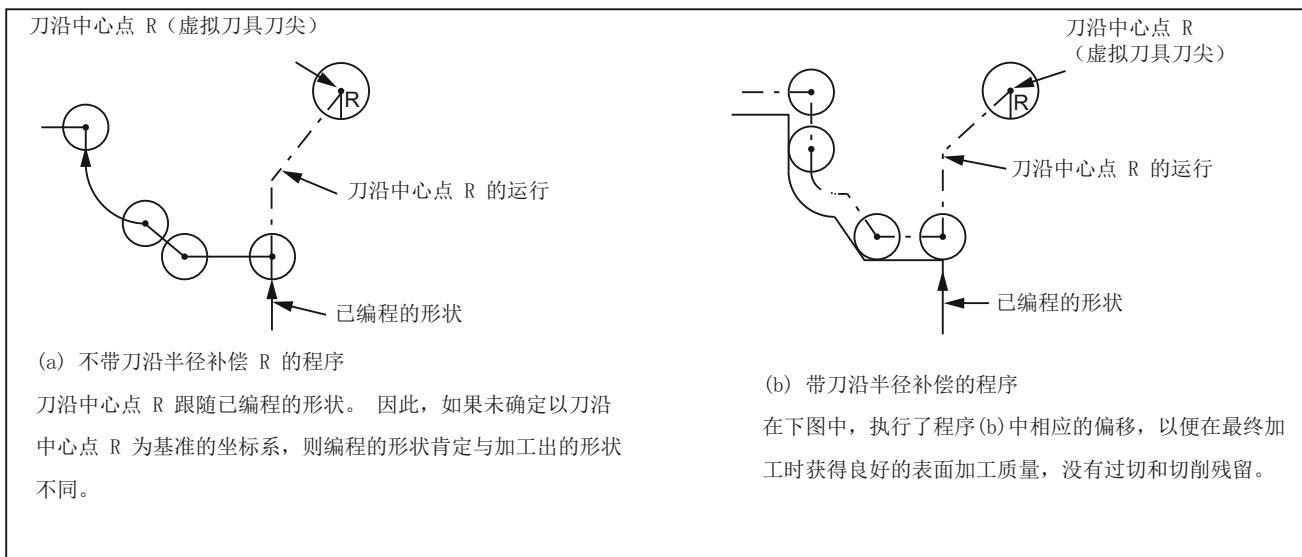


图 4-9 控制点 0 至 9 的程序和刀具运行

### 选择/取消刀沿半径补偿

- 选择刀具补偿  
通过 T 指令选择刀具补偿。
- 启动刀沿半径补偿  
使用下列 G 功能启动和关闭刀沿半径补偿。

表格 4-3 用于启动或关闭刀沿半径补偿的 G 功能

G 功能	功能	G 功能组
G40	取消刀具半径补偿	07
G41	刀具半径补偿（刀具在轮廓的左侧沿加工方向加工）	07
G42	刀具半径补偿（刀具在轮廓的右侧沿加工方向加工）	07

指令 G40 和 G41/G42 是 G 功能组 07 的模态 G 功能。这些指令一直保持有效，直至写入该 G 功能组的其他功能。G40 在上电或 NCK 复位后启用。

通过 G41/G42 和一个 T 指令调用刀沿半径补偿。

## 测量指令

### 4.4 刀具补偿功能

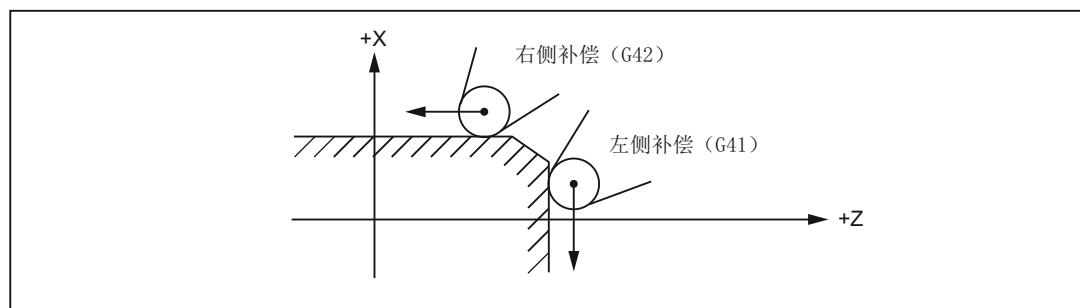


图 4-10 根据加工方向确定刀沿半径补偿

### 补偿方向的转换

补偿方向可在 G41 和 G42 间转换，且不需取消 G40。带旧补偿方向的最后程序段以终点处刀具矢量的正常位置结束。然后按新的补偿方向开始进行补偿（在起点处以正常状态）。

### 刀具半径补偿时的运行轮廓

下图显示了刀具半径补偿的执行。

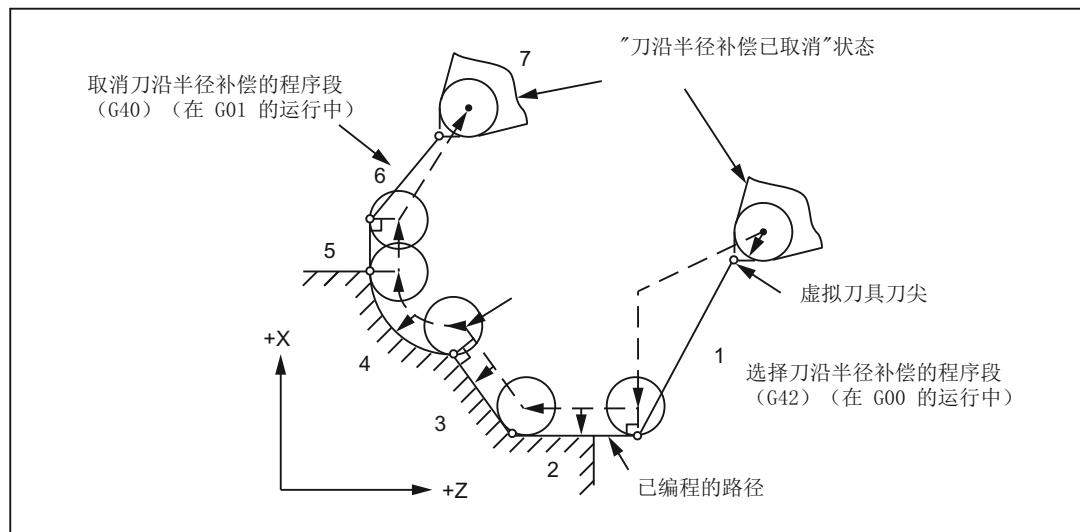


图 4-11 刀具半径补偿的运行轮廓 (G42, 控制点 3)

- 选择（程序段 1）和取消（程序段 6）刀沿半径补偿时，执行补偿运行。因此在选择和取消刀具补偿时必须注意，不会产生轮廓碰撞。

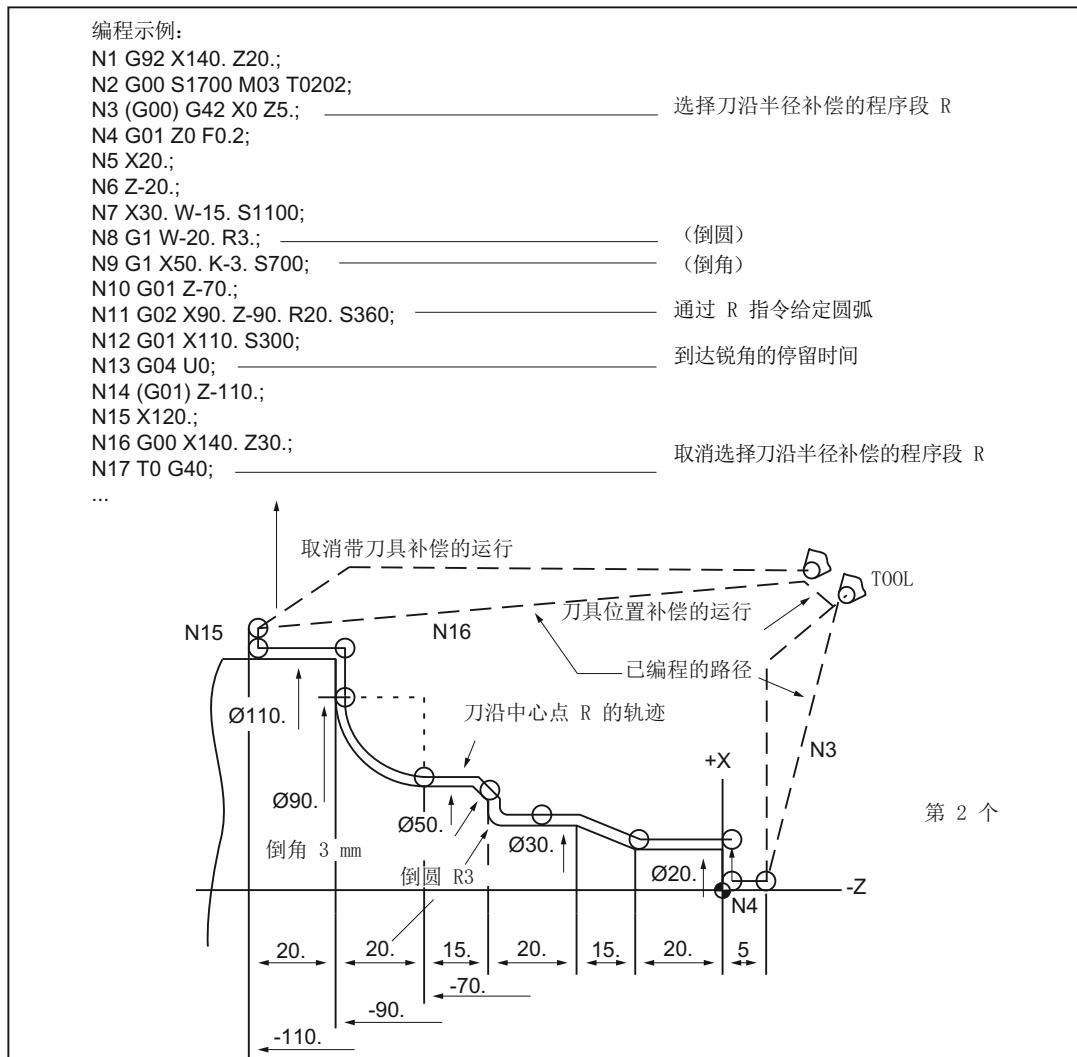


图 4-12 编程示例

## 4.5 S 功能, T 功能, M 功能和 B 功能

### 4.5.1 主轴功能 (S 功能)

通过地址 S 可以给定主轴转速, 单位转每分钟。通过 M3 和 M4 可以选择主轴旋转方向。  
M3 = 主轴右旋, M4 = 主轴左旋。M5 = 主轴停止。详细信息请参见机床制造商的说明资料。

- S 指令模态生效, 即: 写入该指令后指令保持生效, 直至下一个 S 指令。如果通过 M05 停止主轴, S 指令仍保留。如果随后写入 M03 或 M04 而没有给定 S 指令, 则主轴以初始写入的转速启动。
- 如果修改了主轴转速, 请注意主轴当前设定的变速级。详细信息请参见机床制造商的说明资料。
- S 指令的下限 (S0 或接近 S0 的 S 指令) 受驱动电机和主轴的驱动系统的影响; 不同的机床上转速下限也不同。不允许给定 S 负值! 详细信息请参见机床制造商的说明资料。

### 4.5.2 恒定切削速度 (G96, G97)

通过以下 G 功能选择和取消恒定切削速度。指令 G96 和 G97 模态有效并从属于 G 功能组 02。

表格 4-4 用于控制恒定切削速度的 G 指令

G 功能	功能	G 功能组
G96	恒定切削速度“开”	02
G97	取消恒定切削速度	02

### 恒定切削速度“开”(G96)

通过“G96 S...”修改主轴转速（取决于相关的工件直径），使刀沿上的切削速度 **S**（单位：米/分钟或英尺/分钟）保持恒定。

**G96** 启动后，**X** 轴的值作为直径用来监控当前切削速度。如果 **X** 轴的位置改变，主轴转速也随之变化，以保持编程的切削速度。

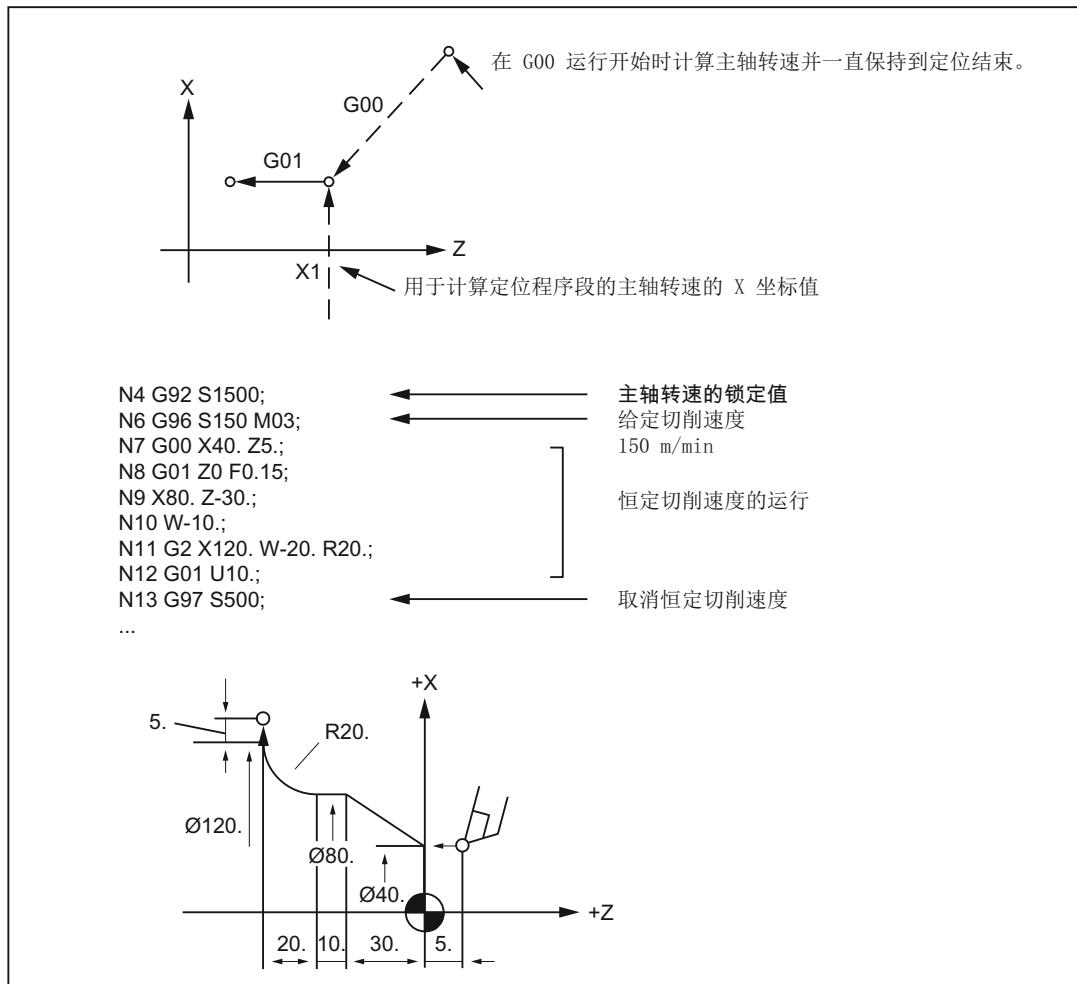


图 4-13 恒定切削速度

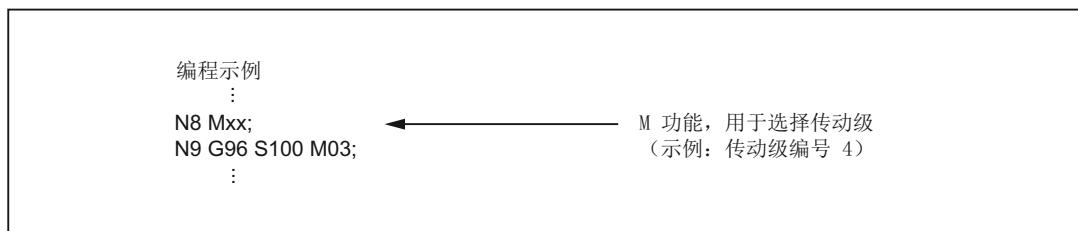
### 取消恒定切削速度 (G97)

在 G97 后，数控系统将一个 **S** 字重新作为主轴转速，单位是转/分钟。如果没有设定新的主轴转速，则保留最后用 **G96** 达到的转速。

### 4.5 S 功能, T 功能, M 功能和 B 功能

#### 选择主轴转速传动级

在可通过 M 指令切换传动级的机床上，在设定 G96 前必须写入用来选择相应传动级的 M 指令。详细信息请参见机床制造商文献。



#### 4.5.3 用 T 指令 (T 功能) 换刀

通过编程 T 字可以直接换刀。

通过机床数据确定 T 功能的生效性能。请注意机床制造商的设计说明。

#### 4.5.4 附加功能 (M 功能)

使用 M 功能可以在机床上控制一些开关操作，比如“冷却液开/关”和其它的机床功能。一些 M 功能已经由数控系统制造商作为固定功能占用（参见下文）。

编程

M... 允许值：0 到 9999 9999（最大整数值），整数

所有空的 M 功能编号都可以由机床制造商预设，例如用于控制夹紧装置或用来启用/关闭其他机床功能的开关功能。参见机床制造商的说明。

下面对 NC 专用的 M 功能进行说明。

#### 用于停止操作的 M 功能(M00, M01, M02, M30)

借助该 M 功能可以释放程序停止，中断或结束加工。此时主轴是否也停止取决于机床制造商的设置。详细信息请参见机床制造商的说明资料。

#### M00 (程序停止)

程序段中带 M00 时加工停止。现在可以进行排屑、再次测量等。并向 PLC 传送一个信号。通过<循环启动>键可以继续程序。

## M01 (可选停止)

M01 可以通过下面的方法设定

- HMI/对话框“程序控制”或者
- VDI 接口

只有当 VDI 接口相应的信号置位或者在 HMI/对话框中选择了“程序控制”时, NC 程序处理才被 M01 停止。

## M30 或 M02 (程序结束)

通过 M30 或 M02 结束程序。

---

### 说明

通过 M00、M01、M02 或 M30 向 PLC 发送信号。

---

### 说明

关于是否通过指令 M00、M01、M02 或 M30 停止主轴或中断冷却液流入的说明请参见机床制造商的资料。

---

## 4.5.5 用于控制主轴的 M 功能

表格 4-5 用于控制主轴的 M 功能

M 功能	功能
M19	定位主轴
M29	切换轴/控制运行中的主轴

借助 M19 主轴运行到设定数据 43240 \$SA\_M19\_SPOS[主轴号]定义的主轴位置上。定位模式保存在 \$SA\_M19\_SPOS 中。

切换主轴运行(M29)的 M 功能号也可以通过机床数据设定。使用 MD20095 \$MC\_EXTERN\_RIGID\_TAPPING\_N\_NR 预设 M 功能号。但只能预设非标准 M 功能的 M 功能号。如不允许预设 M0、M5、M30、M98、M99 等。

在 ISO 模式下, 使用 M29 将主轴切换到进给轴模式。

#### 4.5.6 用于调用子程序的 M 功能

表格 4- 6 用于调用子程序的 M 功能

M 功能	功能
M98	子程序调用
M99	子程序结束

#### 4.5.7 通过 M 功能调用宏

通过 M 号调用子程序（宏）的方式和 G65 类似。

通过机床数据 10814 \$MN\_EXTERN\_M\_NO\_MAC\_CYCLE 和机床数据 10815 \$MN\_EXTERN\_M\_NO\_MAC\_CYCLE\_NAME 可以最多设计 10 个 M 功能替换。

编程方式和 G65 相同。通过地址 L 可以写入重复。

#### 限制

每个零件程序行只可以执行一个 M 功能替换（或只一个子程序调用）。和其他子程序调用发生冲突时，输出报警 12722。在一个已经被替代的子程序中不再执行其它 M 功能替代。

此外，G65 中的限制同样生效。

和预定义的 M 号或和其它定义的 M 号发生冲突时，输出报警，不执行操作。

## 设计示例

通过 M 功能 M101 调用子程序 M101\_MAKRO:

```
$MN_EXTERN_M_NO_MAC_CYCLE[0] = 101
$MN_EXTERN_M_NO_MAC_CYCLE_NAME[0] = "M101_MAKRO"
```

通过 M 功能 M6 调用子程序 M6\_MAKRO:

```
$MN_EXTERN_M_NO_MAC_CYCLE[1] = 6
$MN_EXTERN_M_NO_MAC_CYCLE_NAME[1] = "M6_MAKRO"
```

使用 M 功能换刀的编程示例

```
PROC MAIN
...
N10          M6 X10 V20           ; 调用程序 M6_MAKRO
...
N90          M30
PROC M6_MAKRO
...
N0010        R10 = R10 + 11.11
N0020        IF $C_X_PROG == 1 GOTO N40      ; ($C_X_PROG)
N0030        SETAL(61000)                   ; 没有正确
                                            ; 传送写入的变量
N0040        IF $C_V == 20 GTO N60      ; ($C_V)
N0050        SETAL(61001)
N0060        M17
```

#### 4.5.8 M 功能

##### 通用 M 功能

非专用的 M 功能由机床制造商确定。下面为您列举了使用通用 M 功能的典型示例。详细信息请参见机床制造商的说明资料。如果在同一个程序段中同时写入 M 指令和轴运行指令，视机床制造商的机床数据设置而定，M 功能会在程序段段首执行或在段尾达到轴位置后执行。详细信息请参见机床制造商的说明资料。

表格 4-7 其它通用 M 功能

M 功能	功能	备注
M08	冷却液 开	该 M 功能由机床制造商确定。
M09	冷却液 关	

##### 在一个程序段中给定多个 M 功能

在一个程序段中可以最多写入五个 M 功能。M 功能的组合和限制请参见机床制造商的资料。

##### 附加辅助功能 (B 功能)

B 不用作轴标识符时可用作附加的辅助功能。B 功能作为辅助功能传送给 PLC (H 功能的地址扩展为 H1=)。

示例：B1234 作为 H1=1234 输出。

# 其它功能

## 5.1 程序支持功能

### 5.1.1 固定循环

固定循环可以减轻编程人员在创建程序时的工作量。可以使用 G 功能执行频繁出现的加工步骤；没有固定循环时，必须对多个数控系统程序段进行编程。而通过固定循环可以简化加工程序，并节省存储空间。

在 ISO 编程指令中调用 shell 循环，该循环使用西门子标准循环的功能。这样数控系统程序段中编程的地址便通过系统变量传送给 shell 循环。Shell 循环对这些数据进行匹配，并调用西门子标准循环。

使用了固定循环后，若要继续使用顺时针循环，必须先通过 G80 或 G 代码组 1 中的某个 G 代码将固定循环取消。

#### 轴向切削循环

##### 格式

G.. X... Z... F... ;

G 代码系统 A	G 代码系统 B	G 代码系统 C
G90	G77	G20

通过指令“G... X(U)... Z(W)... F... ;”，按照 1-4 的顺序执行轴向切削循环。

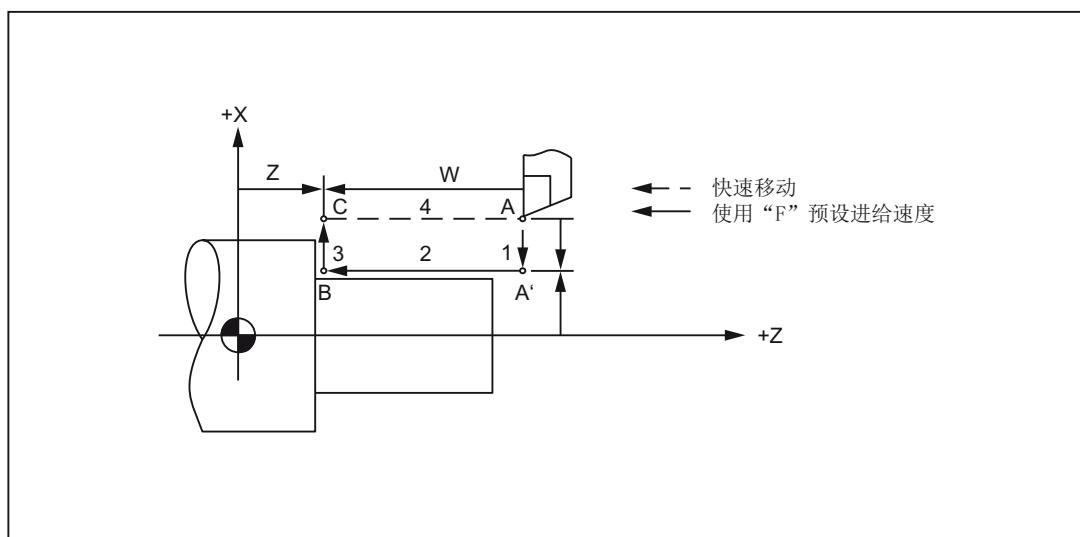


图 5-1 轴向切削循环

## 5.1 程序支持功能

因为 G90 (G77, G20) 是一个模态 G 功能，因此在之后的程序段中只设定 X 轴方向上的进给运行，从而在循环内执行加工。

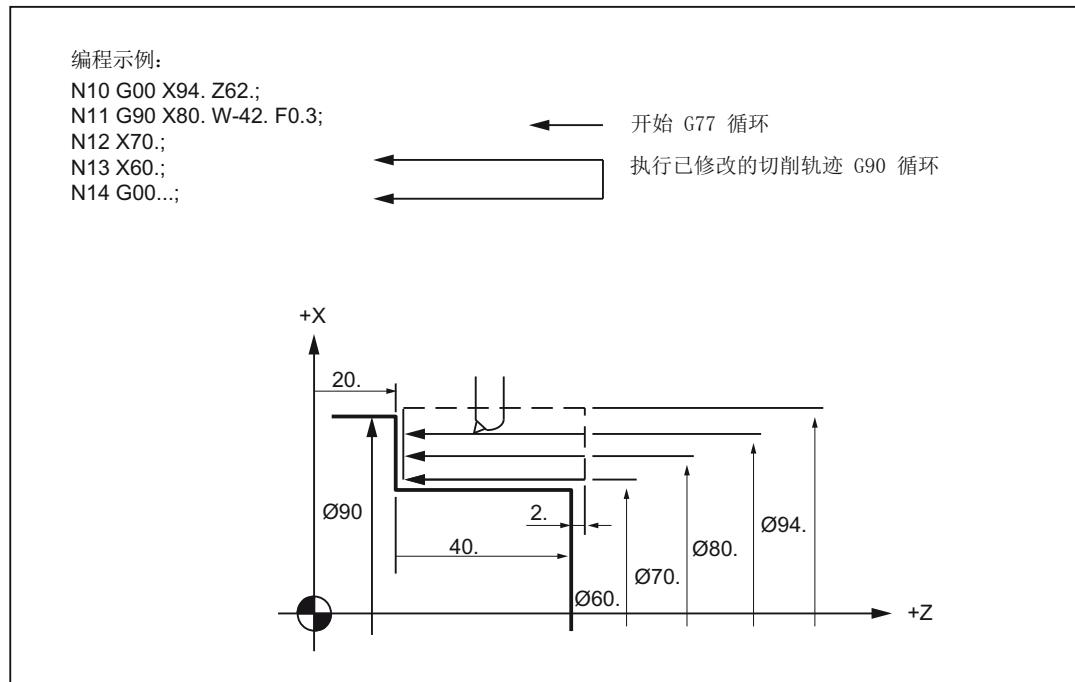


图 5-2 轴向切削循环 (G 代码系统 A)

## 纵向圆锥车削循环

## 格式

G... X... Z... R... F... ;

G 代码系统 A	G 代码系统 B	G 代码系统 C
G90	G77	G20

根据下图，通过指令“G... X(U)... Z(W)... R... F... ;”按照 1-4 的顺序执行纵向圆锥车削循环。

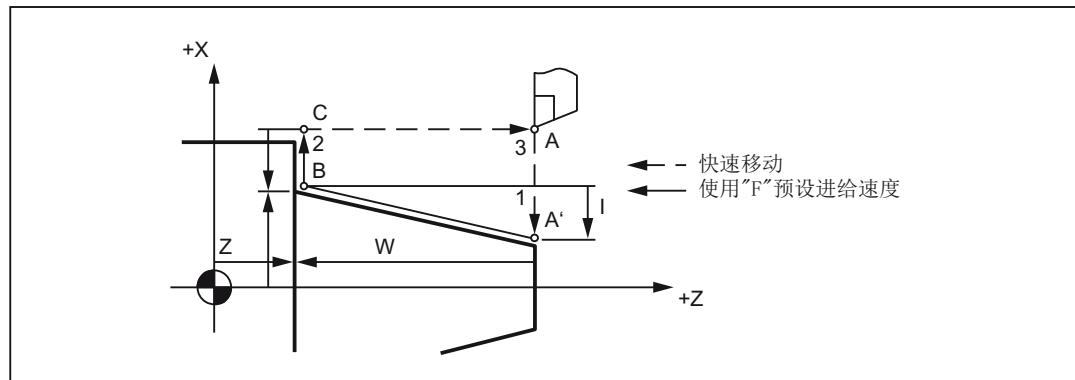


图 5-3 纵向圆锥车削循环

地址符 R 的符号由点 A 决定，观察方向的符号由点 B 决定。

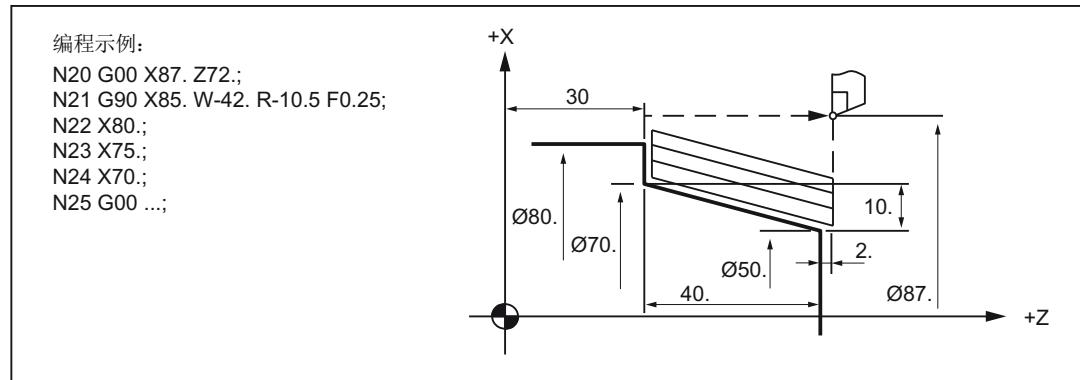
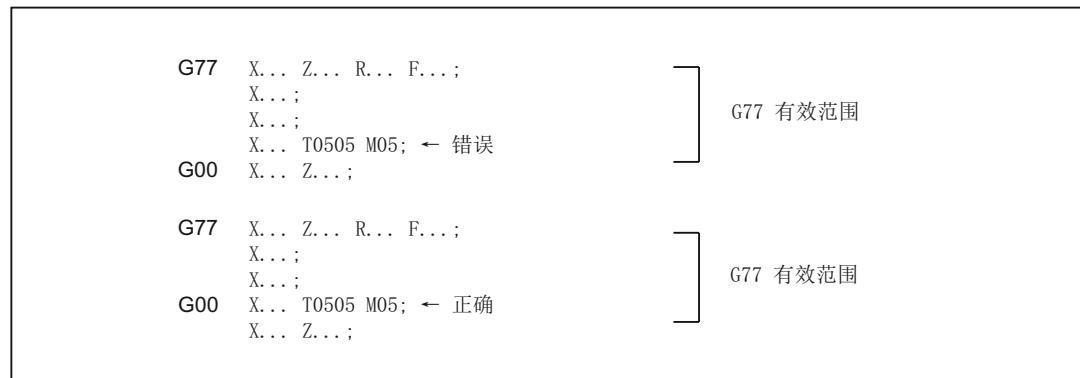


图 5-4 纵向圆锥车削循环 (G 代码系统 A)

- 单程序段运行激活时执行 G90 (G77, G20) 循环，循环虽然不会中途中断，但是会在循环的 1-4 步骤结束后停止。
- 在 G90 (G77, G20) 程序段前，在程序段中设定功能 S, T 和 M，作为执行 G90 (G77, G20) 的切削条件。若未运行轴就在一个程序段中对这些功能进行了设定，那么仅当此程序段设定在 G90 (G77, G20) 运行范围内时，这些功能才生效。



G90 (G77, G20) 运行一直保持有效，直到在一个程序段中对 G 功能组 01 的一个 G 功能重新进行了设定。

## 其它功能

### 5.1 程序支持功能

#### 螺纹切削循环

有四种用于螺纹切削加工的螺纹切削循环：两种循环用于圆柱螺纹切削，两种用于圆锥螺纹切削。

#### 格式

G... X... Z... F... Q... ;

G 代码系统 A	G 代码系统 B	G 代码系统 C
G92	G78	G21

#### 圆柱螺纹切削循环

G... X(U)... Z(W)... F...;  
\_\_\_\_\_ 给定螺距 (L)

通过上述的指令，在下图中按照 1 至 4 的顺序执行圆柱螺纹切削循环。

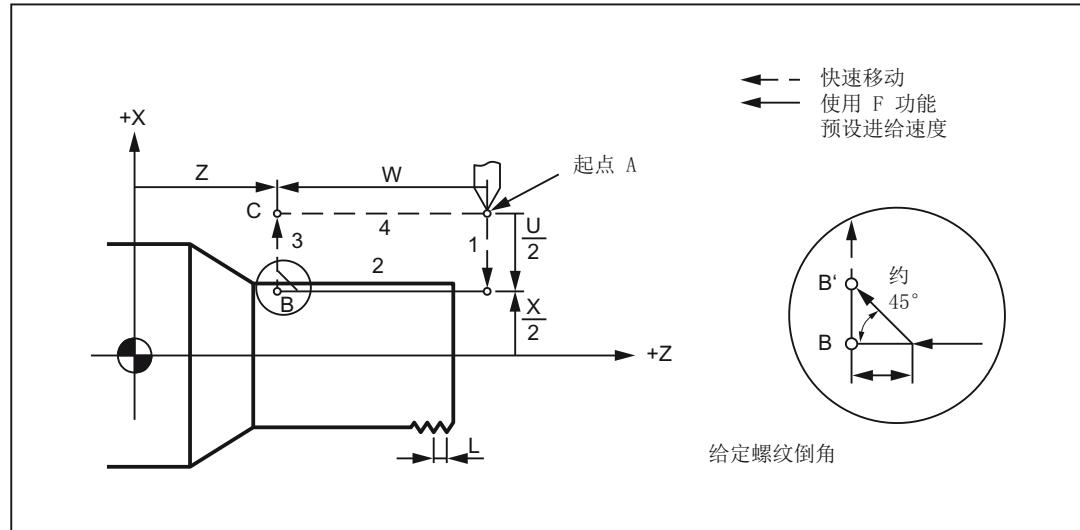


图 5-5 圆柱螺纹切削循环

因为 G92 (G78, G21)是一个模态 G 功能，因此在之后的程序段中只设定 X 轴方向上的切削深度，来执行螺纹切削循环。在这些程序段中不需要再次设定 G92 (G78, G21)。

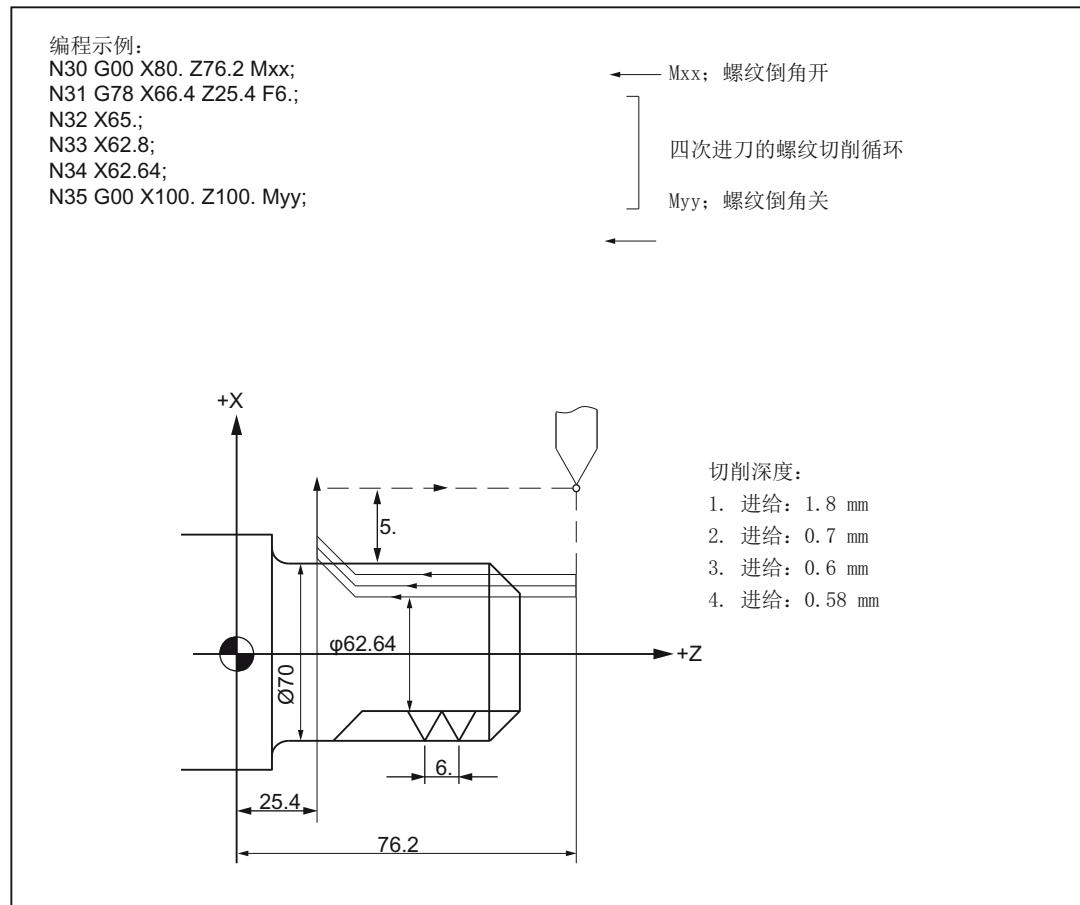


图 5-6 圆柱螺纹切削循环 (G 代码系统 B)

- 当通过激活的单程序段运行执行 G92 (G78, G21)循环时，运行虽然不会在中途中断，但是会在循环的 1-4 几个步骤结束后停止。
- 可在此螺纹切削循环内进行螺纹倒角。通过机床信号触发螺纹倒角。可在 USER DATA, \_ZSFI[26] 中以  $0.1 * L$  为螺距设定螺纹倒角的大小。这里的“L”为所设定的螺距。

## 5.1 程序支持功能

## 圆锥螺纹切削循环

## 格式

G... X... Z... R... F... ;

G 代码系统 A	G 代码系统 B	G 代码系统 C
G92	G78	G21

根据下图，通过指令“G... X(U)... Z(W)... R... F... ;”按照 1-4 的顺序执行圆锥螺纹切削循环。

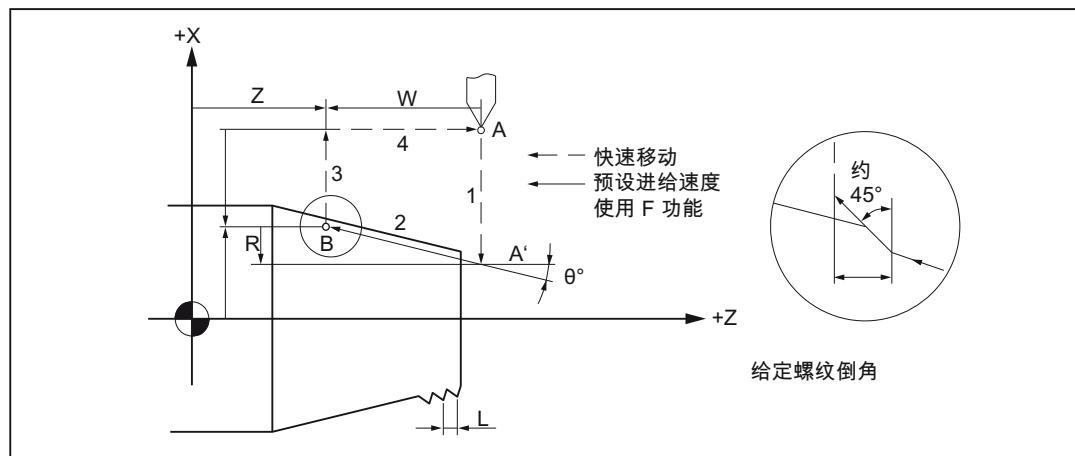


图 5-7 圆锥螺纹切削循环

地址符 **R** 的符号由点 **A** 决定，观察方向的符号由点 **B** 决定。因为 G92 (G78, G21) 是一个模态 **G** 功能，因此在之后的程序段中只设定 **X** 轴方向上的切削深度，来执行螺纹切削循环。在这些程序段中不需要再次设定 G92 (G78, G21)。

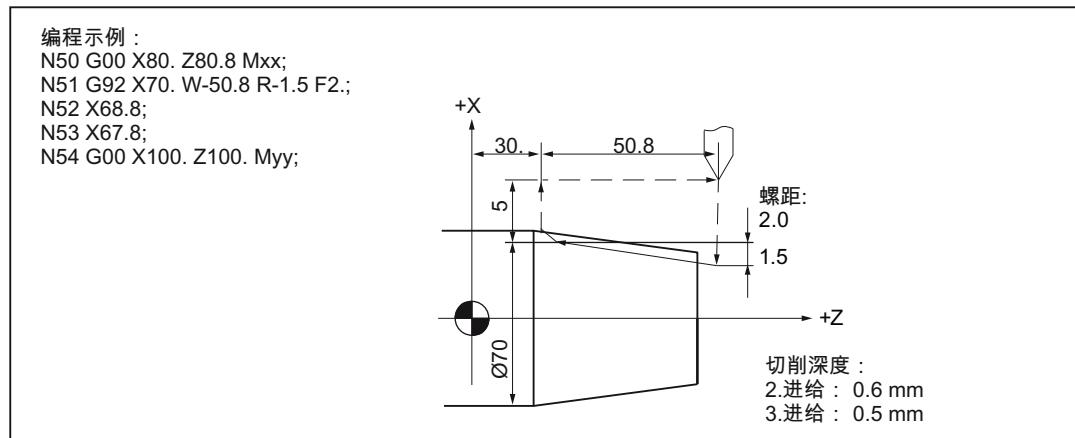


图 5-8 圆锥螺纹切削循环 (G 代码系统 A)

当通过激活的单程序段运行执行 G92 (G78, G21)循环时，运行虽然不会在中途中断，但是会在循环的 1-4 几个步骤结束后停止。

在 G92 (G78, G21) 程序段前，在程序段中设定功能 S, T 和 M，作为执行 G92 (G78, G21) 的切削条件。若未运行轴就在一个程序段中对这些功能进行了设定，那么仅当此程序段设定在 G92 (G78, G21) 运行范围内时，这些功能才生效。

如果在切削刀具位于起点 A 或倒角终点 B 时按下了 <循环启动> 键，将从头开始执行中断的循环。

未选择设置“保持螺纹切削进给率”的情况下，如果在执行螺纹切削循环的过程中按下了 <进给保持> 键，则螺纹切削循环继续运行。在这种情况下加工状态一直保持，直至在螺纹切削循环结束后对刀具进行复位。

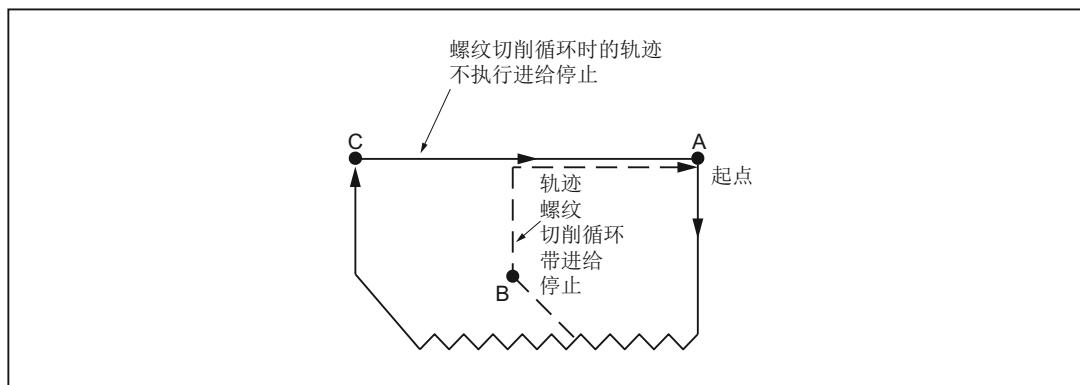


图 5-9 执行螺纹切削循环过程中保持进给率

如果在循环中使用 G92 (G78, G21) 时倒角大小为“0”，会输出报警。

## 5.1 程序支持功能

## 径向切削循环

## 格式

G... X... Z... F... ;

G 代码系统 A	G 代码系统 B	G 代码系统 C
G94	G79	G24

根据下图，通过指令“G... X(U)... Z(W)... F... ;”按照 1-4 的顺序执行横向车削循环。

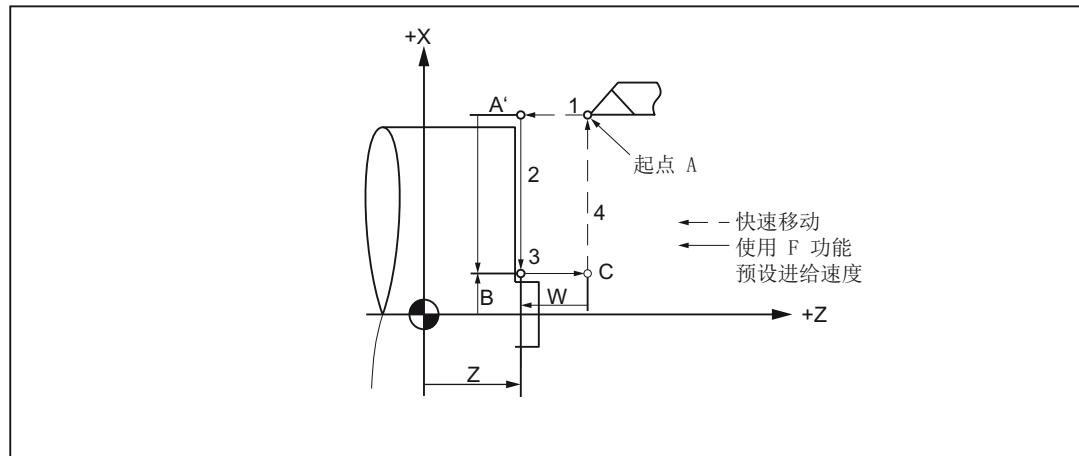


图 5-10 径向切削循环

因为 G94 (G79, G24) 是一个模态 G 功能，因此在之后的程序段中只设定 X 轴方向上的切削深度，来执行螺纹切削循环。在这些程序段中不需要再次设定 G94 (G79, G24)。

编程示例：  
N60 G00 X65. Z42.;  
N61 G79 X20. Z38. F0.35;  
N62 Z34.;  
N63 Z30.;  
N64 G00;

] 3 次 G79 循环加工

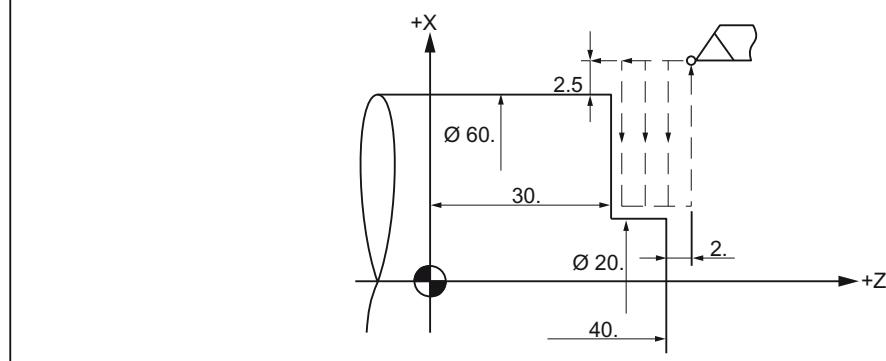


图 5-11 横向车削循环 (G 代码系统 B)

## 横向圆锥车削循环

## 格式

G... X... Z... R... F... ;

G 代码系统 A	G 代码系统 B	G 代码系统 C
G92	G78	G21

根据下图，通过指令“G... X(U)... Z(W)... R... F... ;”按照 1-4 的顺序执行横向圆锥车削循环。

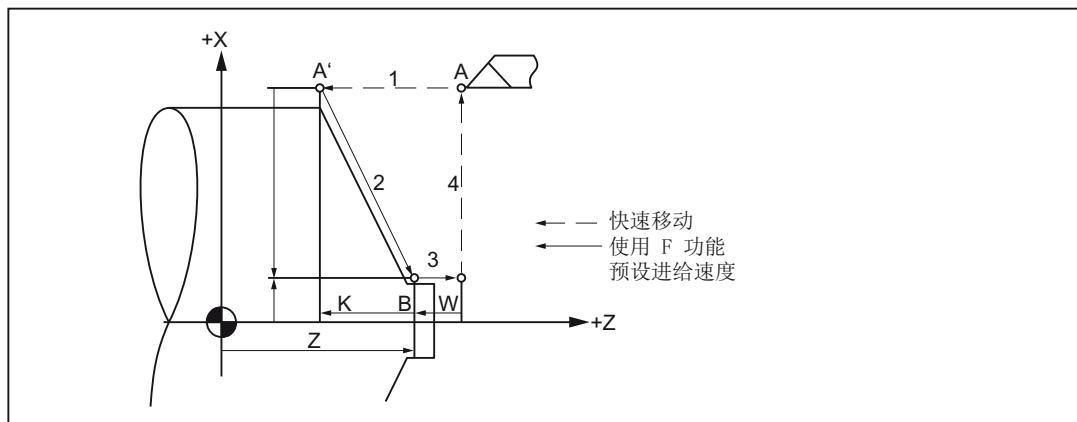


图 5-12 横向圆锥车削循环

地址符 R 的符号由点 A 决定，观察方向的符号由点 B 决定。

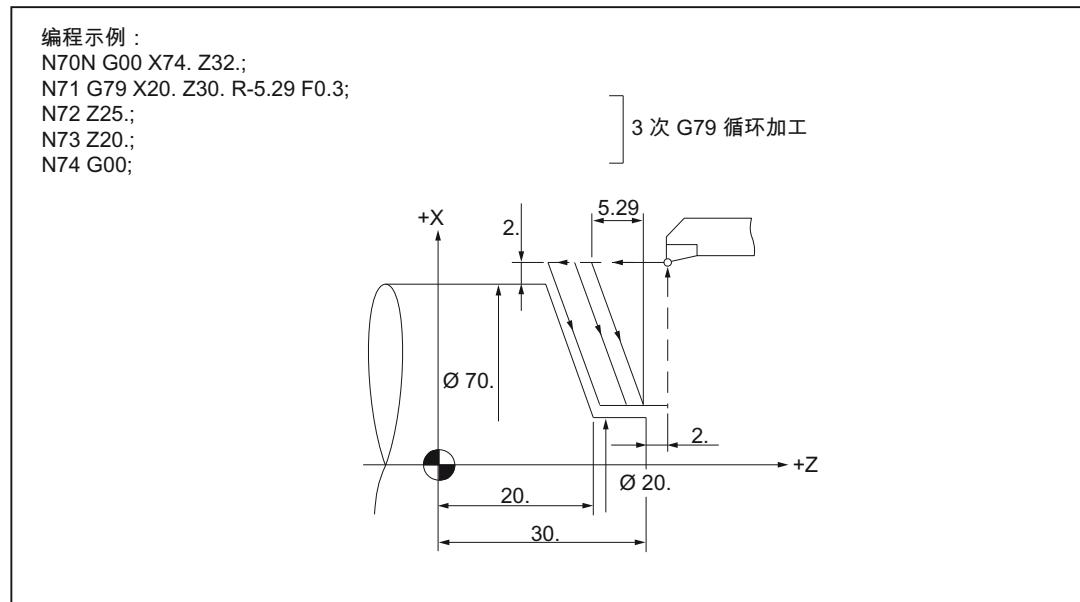


图 5-13 横向圆锥车削循环 (G 代码系统 B)

## 5.1 程序支持功能

在 G94 (G79, G24) 程序段前，在程序段中设定功能 S, T 和 M，作为执行 G94 (G79, G24) 的切削条件。若未运行轴就在一个程序段中对这些功能进行了设定，那么仅当此程序段设定在 G94 (G79, G24) 运行范围内时，这些功能才生效。

单程序段运行激活时执行 G94 (G79, G24) 循环，循环虽然不会中断，但是会在循环的 1-4 步骤结束后停止。

### 5.1.2 多次重复循环

多次重复循环可以减轻编程人员在创建程序时的工作量。可以使用 G 功能执行频繁出现的加工步骤；在没有多次重复循环时，需要对多个数控系统程序段进行编程。而通过多次重复循环可以简化加工程序，并节省存储空间。

在 ISO 编程指令中调用 shell 循环，该循环使用西门子标准循环的功能。这样数控系统程序段中编程的地址便通过系统变量传送给了 shell 循环。Shell 循环对这些数据进行匹配，并调用西门子标准循环。

G 代码系统 A 和 B 中有七种多次重复循环 (G70 至 G76)（参见下表）。请注意，所有这些 G 功能都不是模态有效的。

表格 5-1 车削循环 G70 至 G76 (G 代码系统 A 和 B) 概览

G 代码	说明
G70	精加工循环
G71	轴向粗车循环
G72	径向粗车循环
G73	封闭切削循环
G74	轴向切槽多重循环
G75	径向切槽多重循环
G76	多重螺纹切削循环

G 代码系统 C 中也有此循环。但是其使用的是其它 G 功能。

表格 5-2 车削循环 G72 至 G78 (G 代码系统 C) 概览

G 代码	说明
G72	精加工循环
G73	轴向粗车循环
G74	径向粗车循环
G75	轮廓重复
G76	轴向切槽多重循环
G77	径向切槽多重循环
G78	多重螺纹切削循环

### 说明

下面的循环说明以 G 代码系统 A 和 B 为例。

### 轴向粗车循环(G71)

通过确定最终加工形状等内容就可以便捷的确定粗加工和精加工循环，这表明，通过使用固定循环可以显著减少编程步骤。有两种类型的切削循环。

#### 类型 I

通过  $\Delta d$  (切削进给深度) 按照精加工余量对给定平面进行加工。之后当通过数控系统程序描述轮廓 A 至 A' 至 B 时， $u/2$  和  $\Delta w$  保持不变。

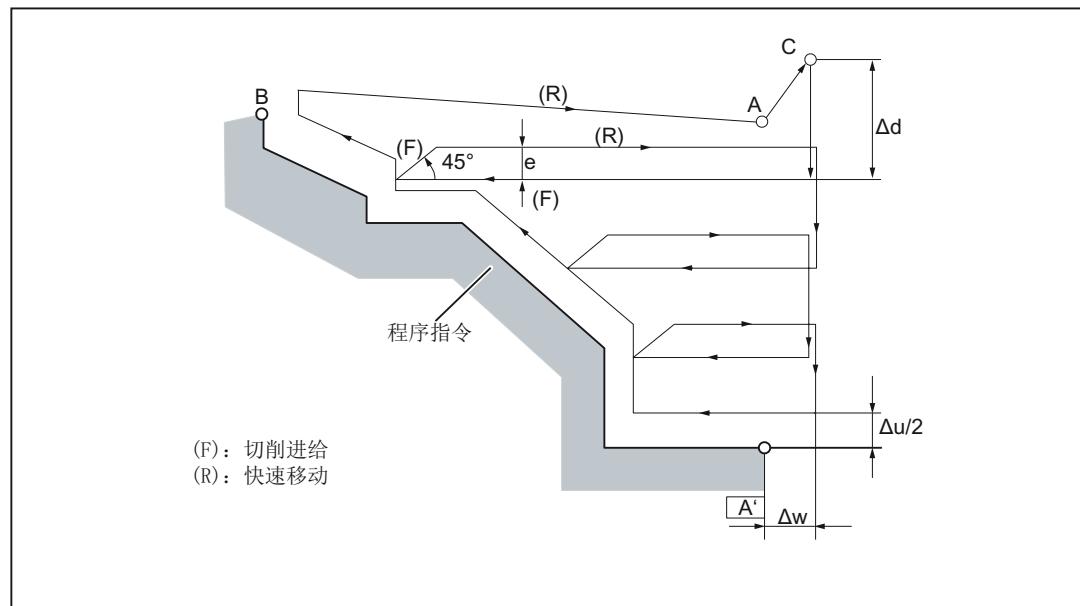


图 5-14 切削循环的切削轨迹，纵向轴

## 5.1 程序支持功能

### 格式

**G71 U... R... ;**

**U:** 切削进给深度 ( $\Delta d$ )，半径编程

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[30] 输入该值，但是会被程序指令的值覆盖。

**R:** (e)，返回量

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[31] 输入该值，但是会被程序指令的值覆盖。

**G71 P... Q... U... W... F... S... T...**

**P:** 用于确定轮廓的起始程序段

**Q:** 用于确定轮廓的结束程序段

**U:** X 轴方向上的精加工余量 ( $\Delta u$ ) (直径/半径编程)

**W:** Z 轴方向上的精加工余量 ( $\Delta w$ )

**F:** 加工进给率

**S:** 主轴转速

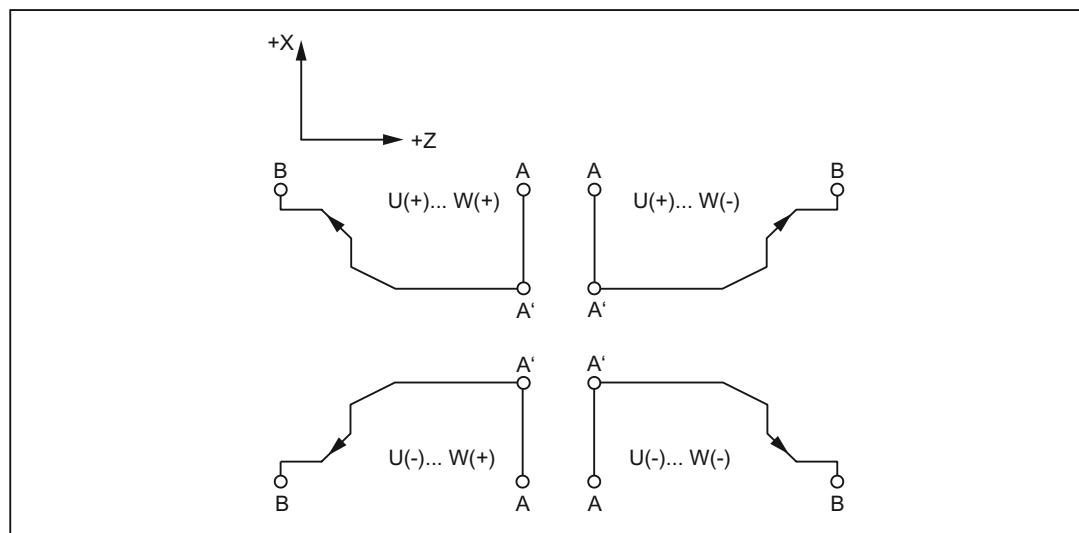
**T:** 刀具选择

忽略数控系统程序段中输出的，通过地址符 P 和 Q 设定的 F 功能，S 功能和 T 功能。只有在 G71 程序段中设定的 F 功能，S 功能和 T 功能才会生效。

### 说明

#### 轴向粗车循环

- 通过地址符 U 设定  $\Delta d$  和  $\Delta u$ 。通过地址符 P 和 Q 设定时， $\Delta u$  就是这种情况。
- 共有四个不同的切削区域。如下图所示， $\Delta u$  和  $\Delta w$  的符号可以不同：



**说明****轴向粗车循环**

- 在通过地址 P 设定的程序段中确定点 A 和 A' 间的轮廓 (G00 或 G01)。在此程序段中不能设定 Z 轴上的运行指令。  
所确定的点 A' 和 B 之间的轮廓在 X 轴和 Z 轴上必须为持续上升，或者持续下降的图形。
- 在通过地址符 P 和 Q 设定的数控系统程序段范围内不能调用子程序。

**类型 II**

与类型 I 相反，在类型 II 中持续的上升或下降不是必需的，也就是说可以出现槽状。

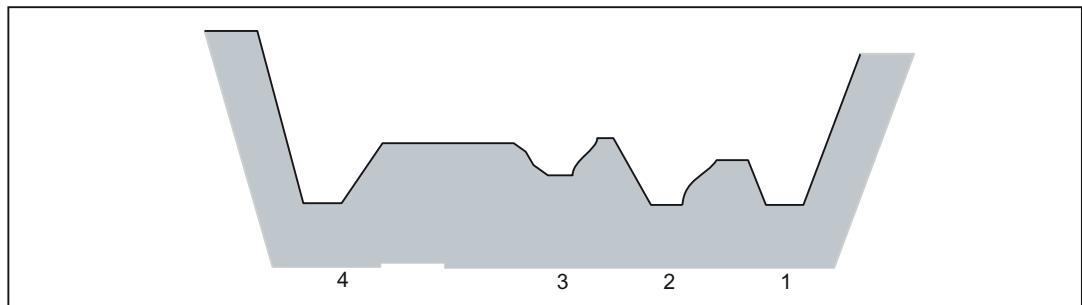


图 5-15 切削循环中的槽 (类型 II)

但是 Z 轴上的轮廓必须同为上升或同为下降。比如，下面的这个轮廓无法进行加工：

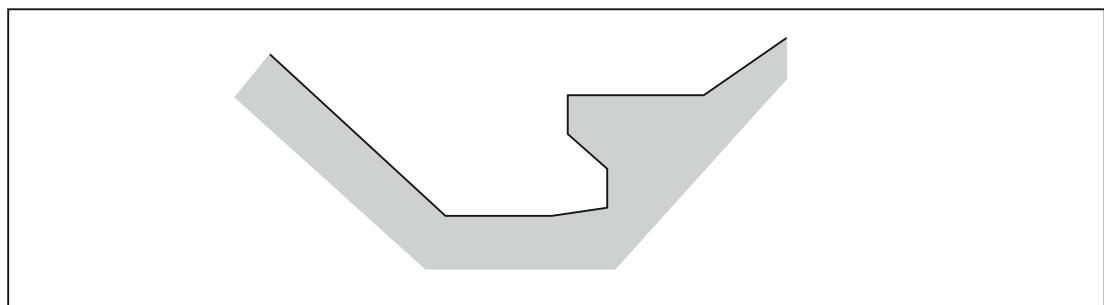


图 5-16 在 G71 循环中无法加工的轮廓

**类型 I 和类型 II 的区别**

**类型 I:** 轮廓描述时，在第一个程序段中只设定一个轴。

**类型 II:** 在轮廓描述的第一个程序段中设定两个轴。

如果需要使用类型 II，但第一个程序段中未设定 Z 轴上的运行，则必须设定 W0。

## 5.1 程序支持功能

## 示例

类型 I	类型 II
G71 U10.0 R4.0 ;	G71 U10.0 R4.0 ;
G71 P50 Q100 .... ;	G71 P50 Q100 ..... ;
N50 X(U)... ;	N50 X(U)... Z(W)... ;
::	::
::	::
N100..... ;	N100..... ;

## 径向粗车循环(G72)

通过 G72 指令可在横向侧按照精加工余量对切削循环进行编程。在通过 G71 调用的循环中，平行于 Z 轴进行加工；与之相对，通过 G72 调用的循环则是平行于 X 轴来进行加工的。通过 G72 调用的循环所执行的加工与通过 G71 调用的相同，只是方向不同。

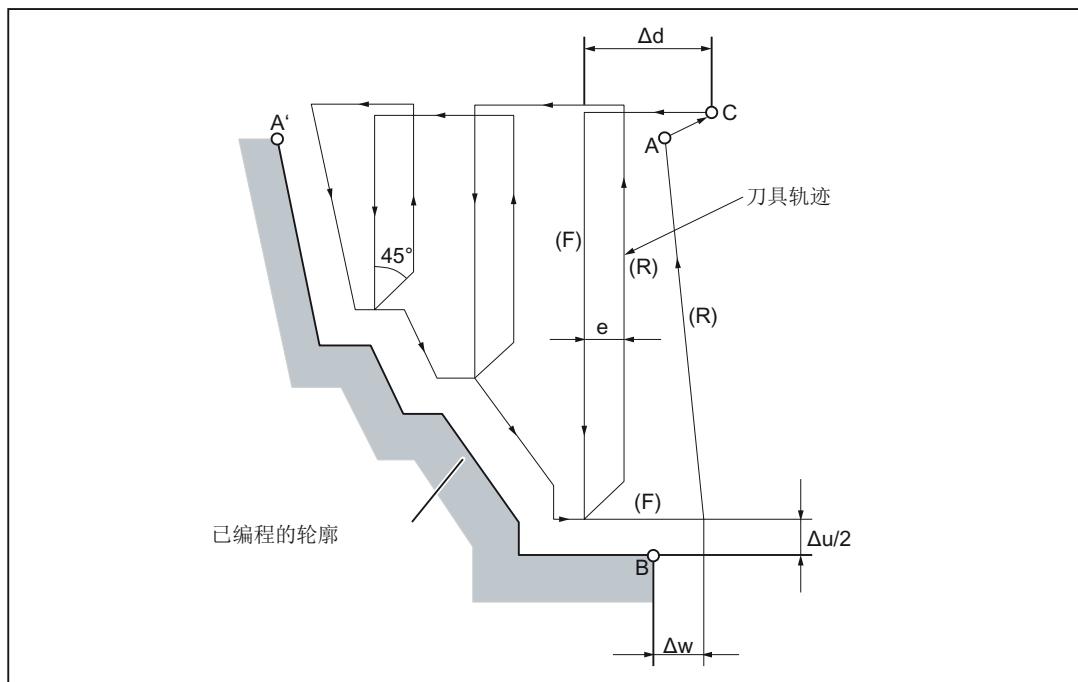


图 5-17 切削循环的切削轨迹，横向轴

## 格式

**G72 W... R... ;**

地址 W ( $\Delta d$ ) 和 R (e) 的含义与 U 和 R 相同。

**G72 P... Q... U... W... F... S... T... ;**

地址 P, Q, U ( $\Delta u$ ), W ( $\Delta w$ ), F, S 和 T 的含义与循环 G71 中的相同。

## 说明

### 径向粗车循环

- 通过地址“U”或“W”来确定值  $\Delta i$  和  $\Delta k$ , 或者  $\Delta u$  和  $\Delta w$ 。它们的含义则通过地址符 P 和 Q 在 G73 程序段中确定。当未在同程序段中设定 P 和 Q 时, 地址符 U 和 W 与  $\Delta i$  或  $\Delta k$  相关联。当未在同程序段中设定 P 和 Q 时, 地址符 U 和 W 与  $\Delta u$  和  $\Delta w$  相关联。
- 共有四个不同的切削区域。如下图所示,  $\Delta u$  和  $\Delta w$  的符号可以不相同:

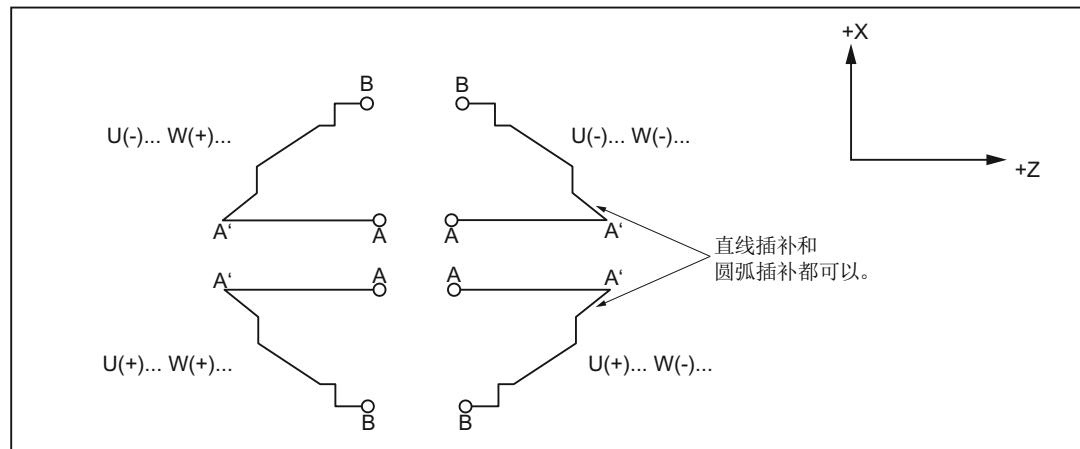


图 5-18 横向切削时, 材料减少量中 U 和 W 的符号

## 说明

### 径向粗车循环

- 通过以地址符 P 设定的程序段 (G00 或 G01) 确定点 A 和 A' 间的轮廓。在此程序段中不能设定 X 轴上的运行指令。所确定的点 A' 和 B 之间的轮廓在 X 轴和 Z 轴上必须为持续上升, 或者持续下降的图形。
- 通过指令 G73 以及 P 和 Q 的设定值在循环内执行加工。接下来对四个切削区域进行观察。请特别注意  $\Delta u$ ,  $\Delta w$ ,  $\Delta k$  和  $\Delta i$  的符号。加工循环结束时, 刀具立即返回点 A。

## 5.1 程序支持功能

### 封闭切削循环 (G73)

当对工件进行加工，其形状与最终加工相似时（比如铸铁工件或者锻造工件），封闭切削循环 G73 会更有效。

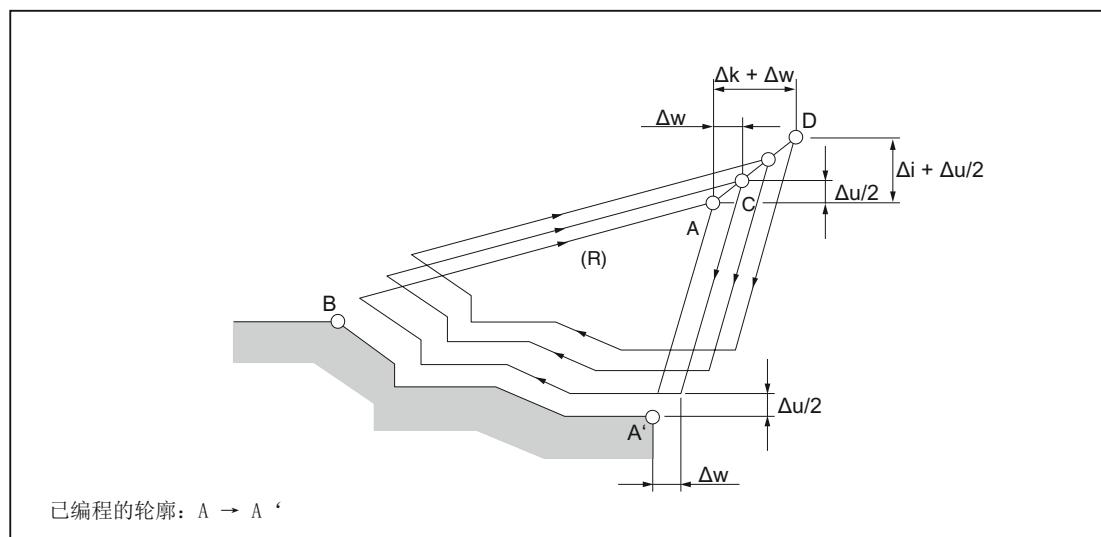


图 5-19 封闭切削循环的切削轨迹

### 格式

**G73 U... W... R... ;**

**U:** (半径编程时) X 轴方向上起点和当前刀具位置之间的距离( $\Delta i$ )。

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[32] 输入该值，但是会被程序指令的值覆盖。

**W:** Z 轴方向上起点和当前刀具位置之间的距离( $\Delta k$ )。

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[33] 输入该值，但是会被程序指令的值覆盖。

**R:** 与轮廓 (d) 平行的切削数量。

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[34] 输入该值，但是会被程序指令的值覆盖。

**G73 P... Q... U... W... F... S... T... ;**

**P:** 用于确定轮廓的起始程序段

**Q:** 用于确定轮廓的结束程序段

**U:** X 轴方向上的精加工余量 ( $\Delta u$ ) (直径/半径编程)

**W:** Z 轴方向上的精加工余量 ( $\Delta w$ )

**F:** 加工进给率

**S:** 主轴转速

**T:** 刀具选择

忽略数控系统程序段中输出的，通过地址符 P 和 Q 设定的 F 功能，S 功能和 T 功能。只有在 G73 程序段中设定的 F 功能，S 功能和 T 功能才会生效。

## 精加工循环 (G70)

执行 G71, G72 或 G73 编写的粗加工时，通过下面的指令进行精加工。

### 格式

**G70 P... Q... ;**

**P:** 用于确定轮廓的起始程序段

**Q:** 用于确定轮廓的结束程序段

### 说明

#### 精加工循环

1. 在程序段间设定的，通过地址符 P 和 Q 确定的功能在 G70 循环中生效，而在 G71, G72 和 G73 程序段中设定的 F 功能，S 功能和 T 功能不生效。
2. G70 加工循环结束时，刀具立即返回起点，并读取下一个程序段。
3. 在通过地址符 P 和 Q 确定的程序段中可以调用子程序。

### 举例

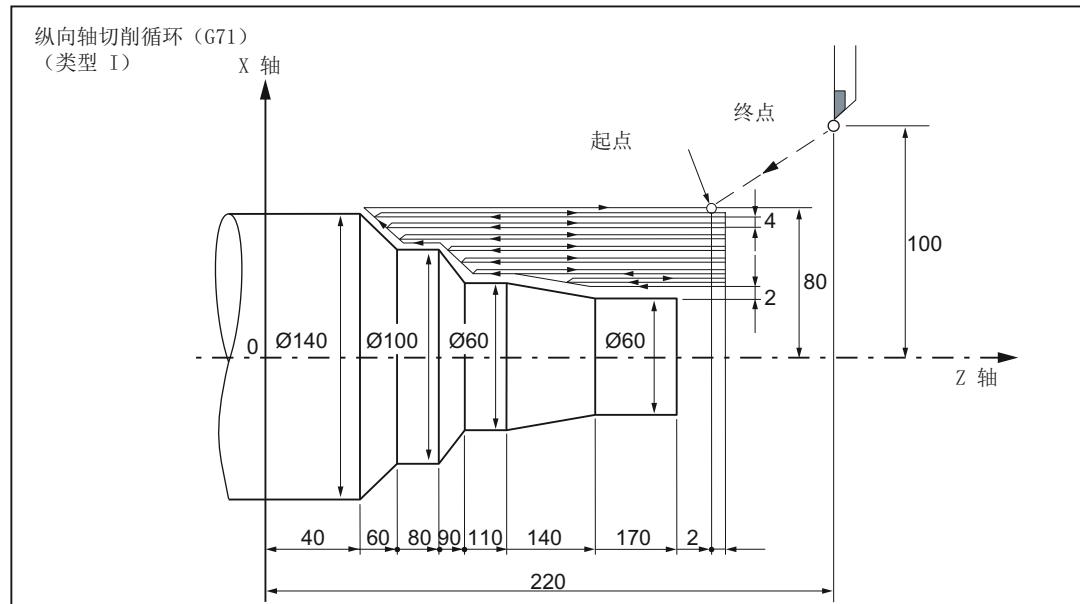


图 5-20 轴向粗车循环

## 5.1 程序支持功能

(直径编程, 公制输入)

```
N010 G00 X200.0 Z220.0  
N011 X142.0 Z171.0  
N012 G71 U4.0 R1.0  
N013 G71 P014 Q020 U4.0 W2.0 F0.3 S550  
N014 G00 X40.0 F0.15 S700  
N015 G01 Z140.0  
N016 X60.0 Z110.0  
N017 Z90.0  
N018 X100.0 Z80.0  
N019 Z60.0  
N020 X140.0 Z40.0  
N021 G70 P014 Q020  
N022 G00 X200 Z220
```

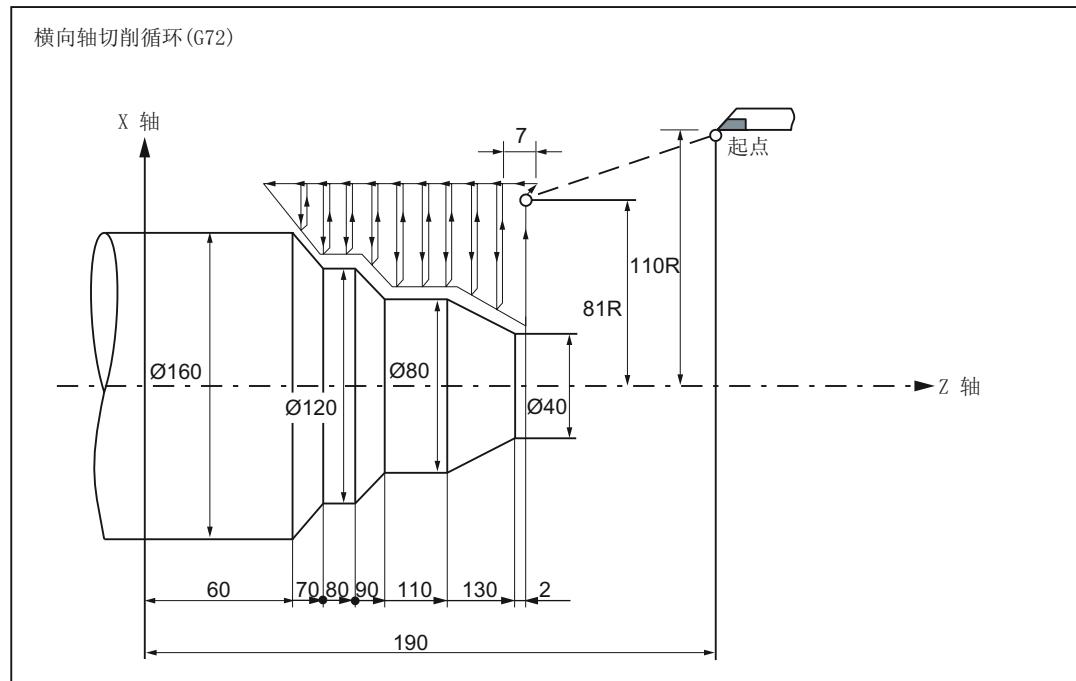


图 5-21 径向粗车循环

(直径编程, 公制输入)

```

N010 G00 X220.0 Z190.0
N011 G00 X162.0 Z132.0
N012 G72 W7.0 R1.0
N013 G72 P014 Q019 U4.0 W2.0 F0.3
N014 G00 Z59.5 F0.15 S200
N015 G01 X120.0 Z70.0
N016 Z80.0
N017 X80.0 Z90.0
N018 Z110.0
N019 X36.0 Z132.0
N020 G70 P014 Q019
N021 X220.0 Z190.0

```

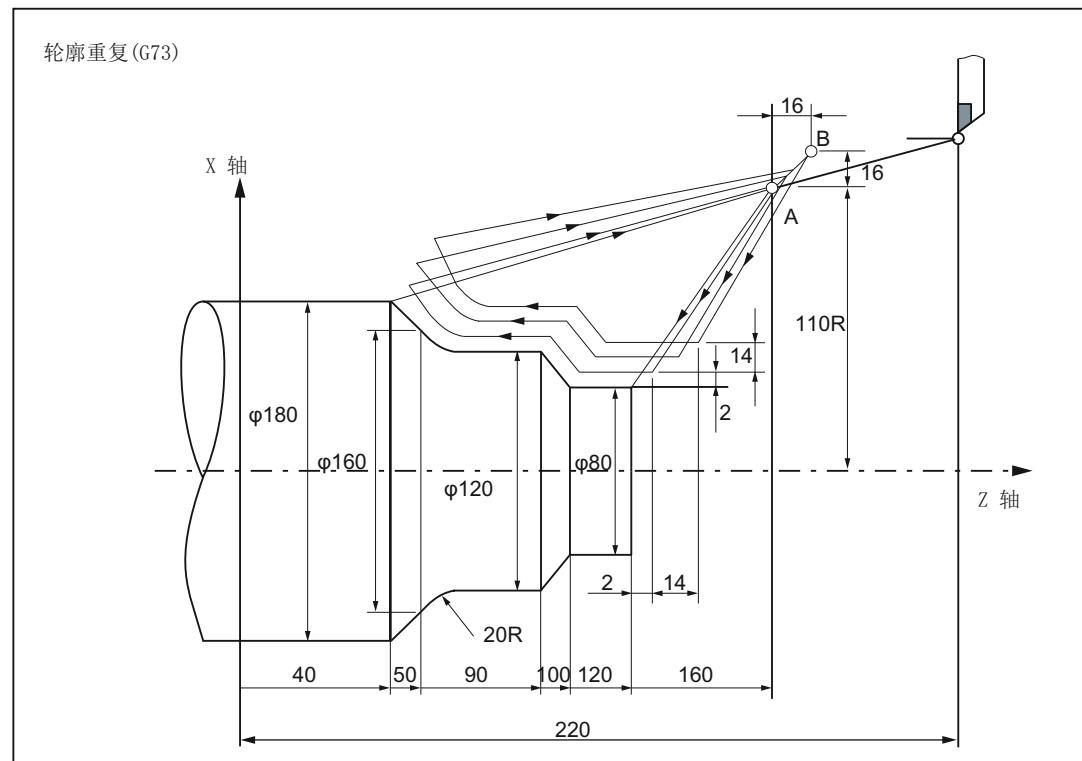


图 5-22 轮廓重复

## 5.1 程序支持功能

(直径编程, 公制输入)

```
N010 G00 X260.0 Z220.0  
N011 G00 X220.0 Z160.0  
N012 G73 U14.0 W14.0 R3  
N013 G73 P014 Q020 U4.0 W2.0 F0.3 S0180  
N014 G00 X80.0 Z120.0  
N015 G01 Z100.0 F0.15  
N017 X120 Z90.0  
N018 Z70  
N019 G02 X160.0 Z50.0 R20.0  
N020 G01 X180.0 Z40.0 F0.25  
N021 G70 P014 Q020  
N022 G00 X260.0 Z220.0
```

## 轴向切槽多重循环 (G74)

在 G74 调用的循环中, 平行于 Z 轴进行断屑加工。

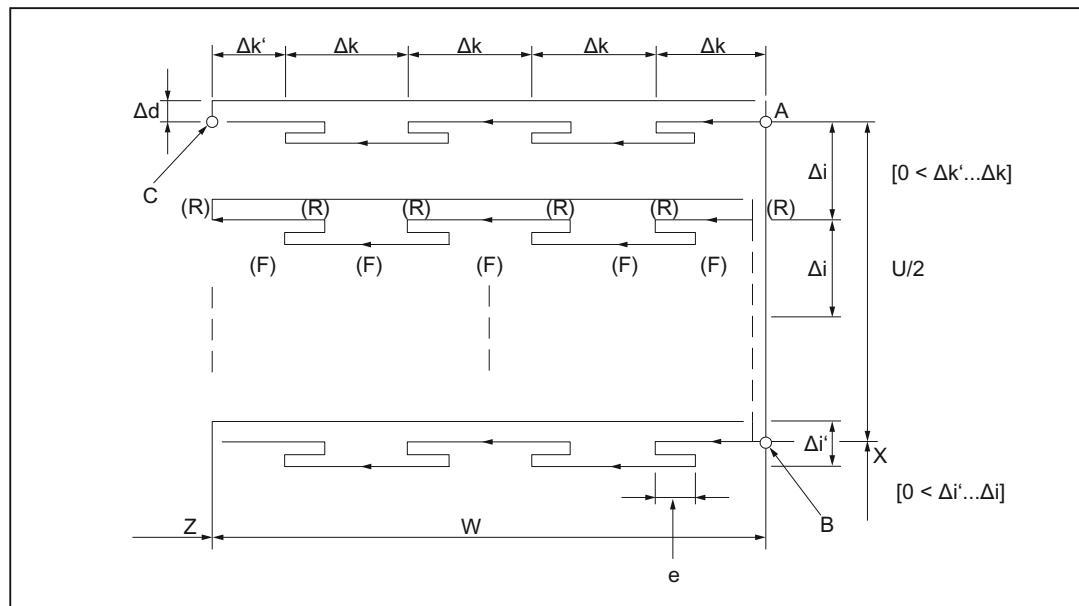


图 5-23 深孔钻削循环的切削轨迹

## 格式

**G74 R... ;**

**R:** d), 返回量

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[29] 输入该值，但是会被程序指令的值覆盖。

**G74 X(U)... Z(W)... P... Q... R... F...(f) ;**

**X:** 起点 X (位置设定 绝对)

**U:** 起点 X (位置设定 增量)

**Z:** 起点 Z (位置设定 绝对)

**W:** 起点 Z (位置设定 增量)

**P:** X 轴方向上的进给量 ( $\Delta i$ ) (不带符号)

**Q:** Z 轴方向上的进给量 ( $\Delta k$ ) (不带符号)

**R:** 冲孔底上的返回量 ( $\Delta d$ )

**F:** 进给率

---

## 说明

### 轴向切槽多重循环

1. 通过地址 R 确定 “e” 和  $\Delta d$ ，通过设定地址 X (U) 确定“e”和“d”的含义。只在设定了 X (U) 时，才使用  $\Delta d$ 。
  2. 通过指令 G74 和 X (U) 设定的值执行加工循环。
  3. 若循环用于钻孔，则不能使用地址 X (U) 和 P。
-

## 5.1 程序支持功能

### 径向切槽多重循环 (G75)

在 G75 调用的循环中，平行于 X 轴进行断屑加工。

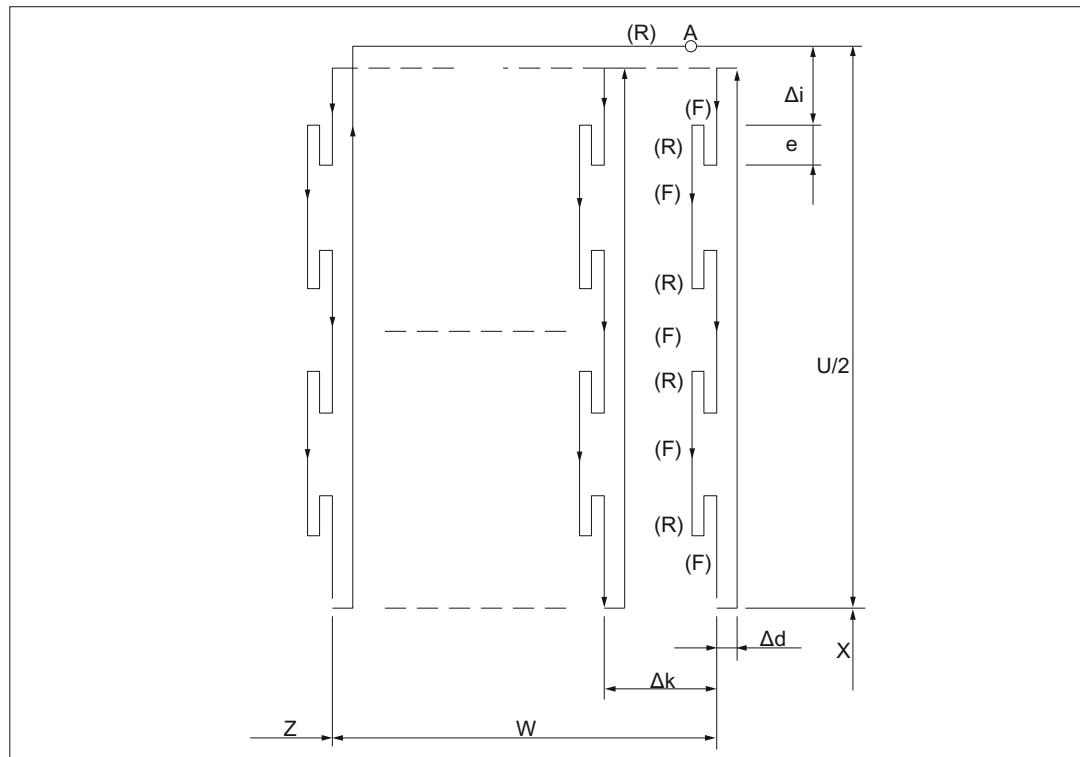


图 5-24 径向切槽多重循环的切削轨迹 (G75)

### 格式

G75 R...;

G75 X(U)... Z(W)... P... Q... R... F...;

地址的含义与循环 G74 中相同。

### 说明

若循环用于钻孔，则不能使用地址 Z (W) 和 Q。

### 多重螺纹切削循环 (G76)

通过 G76 调用一个自动的多重螺纹切削循环，用于切削圆柱螺纹或者圆锥螺纹，在此循环中以一定的螺纹角度进行进给。

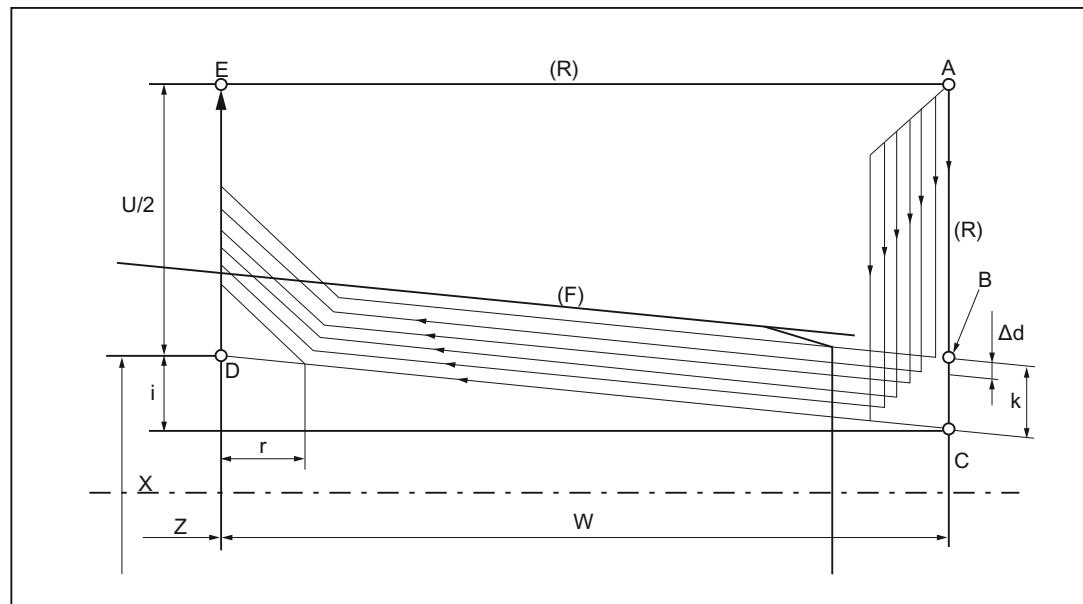


图 5-25 多头螺纹切削循环的切削轨迹

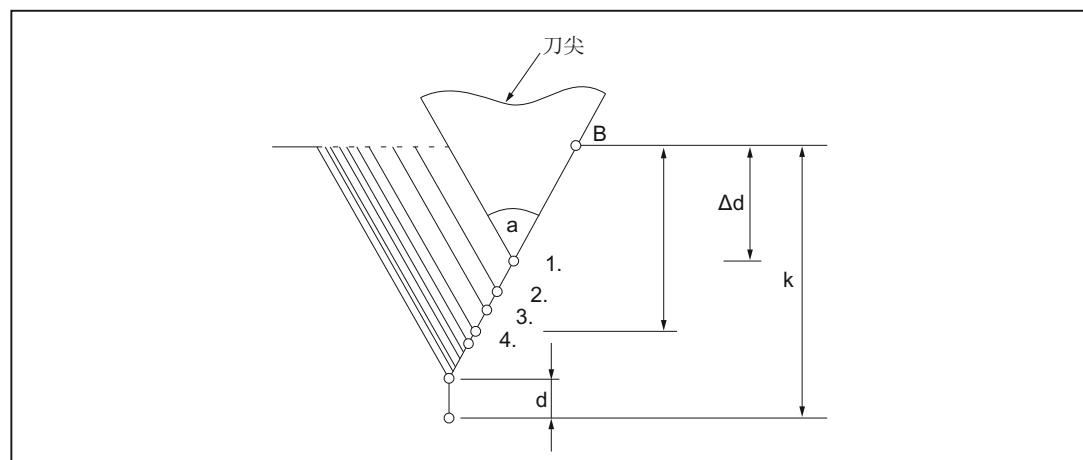


图 5-26 螺纹切削进给

## 5.1 程序支持功能

### 格式

G76 P... (m, r, a) Q... R... ;

P:

m: 精切数量

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[24] 输入该值，但是会被程序指令的值覆盖。

r: 螺纹末端的倒角大小 (1/10 \* 螺距)

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[26] 输入该值，但是会被程序指令的值覆盖。

a: 切削角度

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[25] 输入该值，但是会被程序指令的值覆盖。

上面设定的所有参数会同时通过地址 P 设定。

地址 P 的一个示例：

G76 P012055 Q4 R0.5



Q: 最小进给深度 ( $\Delta d_{min}$ )，半径值

只有在循环加工中的切削深度 ( $\Delta d - \Delta d-1$ ) 小于其极限值时，切削深度才与地址 Q 设定的值保持关联。

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[27] 输入该值，但是会被程序指令的值覆盖。

R: 精加工余量

该值一直为模态有效，直至在程序中写入了其它值。也可通过 USER DATA, \_ZSFI[28] 输入该值，但是会被程序指令的值覆盖。

G76 X(U)... Z(W)... R... P... Q... F... ;

X, U: X 轴方向上的螺纹终点 ((X) 的位置设定为绝对值, (U) 的位置设定为增量值)

Z, W: Z 轴方向上的螺纹终点

R: 圆锥螺纹的半径差(i)。对于简单的圆柱螺纹 i = 0

P: 螺纹深度(k), 半径值

Q: 第 1 切削的进给量( $\Delta d$ ), 半径值

F: 螺纹螺距(L)

**说明****多重螺纹切削循环**

1. 地址符 P, Q 和 R 设定的数据的含义由 X (U) 和 X (W) 确定。
2. 通过指令 G76, 以及 X (U) 和 Z (W) 设定的值执行加工循环。使用该循环时执行“单刃切削”，刀具刀尖负载降低。
  - 将每个循环的切削量归入相应的切削深度，以保持其恒定。 $\Delta d$  归入第一条轨迹， $\Delta d_n$  归入第 n 条轨迹。对应地址符的符号会涉及四个对称的截面。
3. G32 螺纹切削和 G76 螺纹切削循环的提示在这里也适用。

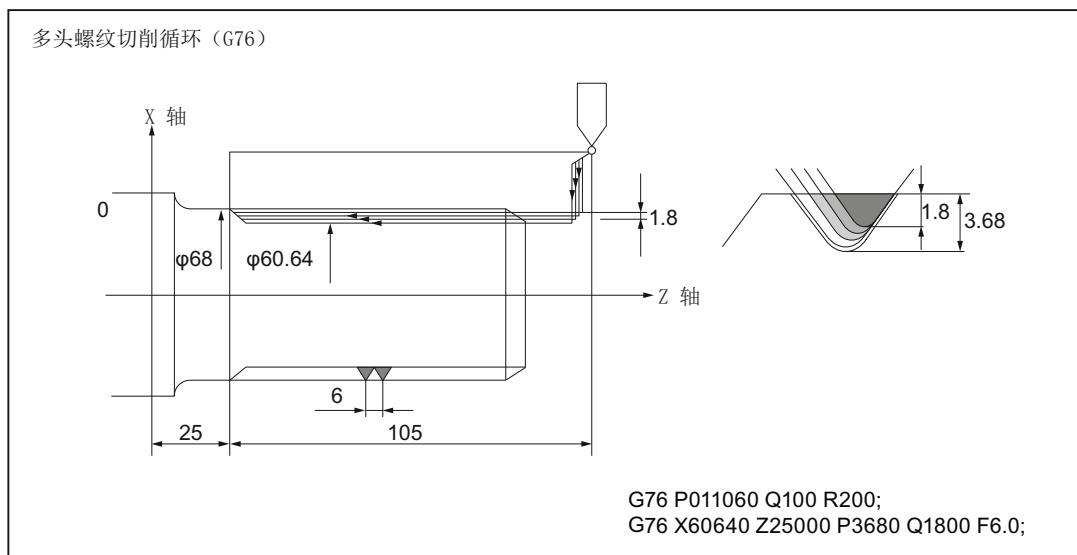
**举例**

图 5-27 螺纹切削循环(G76)

**说明****边界条件**

1. 在运行方式 MDA 中，不允许使用指令 G70, G71, G72 或 G73；否则会输出报警 14011。而 G74, G75 和 G76 则只能在运行方式 MDA 中使用。
2. 在 G70, G71, G72 或 G73 程序段，以及通过地址 P 和 Q 设定的顺序编号中，不允许编程 M98（调用子程序）和 M99（结束子程序）。
3. 在带有地址符 P 和 Q 设定的顺序编号的程序段中不允许编程以下指令：
  - 一次性生效的 G 功能 (G04 暂停时间除外)
  - G 功能组 01 的 G 功能 (G00, G01, G02 和 G03 除外)
  - G 功能组 06 的 G 功能
  - M98/M99
4. G70, G71, G72 和 G73 所定义轮廓的结束运行不能用倒角或者倒圆加工来结束。否则会输出一个故障信息。
5. 在 G74, G75 和 G76 循环中，地址 P 和 Q 使用最小增量输入来设定运行路径和切削深度。
6. 在循环 G71, G72, G73, G74, G75, G76 和 G78 中不能执行刀沿半径补偿。

## 5.1 程序支持功能

### 5.1.3 钻孔循环 (G80 至 G89)

通过钻孔加工固定循环 (G80 至 G89)，可对钻孔加工进行编程，这通常需要在单程序段指令中写入多个指令块。固定循环调用的编程可通过 G80 再次取消。

对于所有 G 代码系统，用于调用固定循环 G80 至 G89 的 G 功能都相同。

#### 用于调用固定循环的 G 功能，固定循环的坐标轴运行模式

下表中给出了用于调用固定循环的 G 功能。

表格 5-3 钻削循环

G 代码	钻孔 (- 方向)	钻孔底上的加工	返回 (+ 方向)	应用
G80	-	-	-	取消选择
G83	中断的切削进给率	-	快速移动	端面深孔钻削
G84	切削进给率	暂停时间 -> 主轴 左转	切削进给率	端面螺纹钻孔
G85	切削进给率	暂停时间	切削进给率	端面钻孔
G87	中断的切削进给率	暂停时间	快速移动	侧面深孔钻削
G88	切削进给率	暂停时间 -> 主轴 左转	切削进给率	侧面螺纹钻孔
G89	切削进给率	暂停时间	切削进给率	侧面钻孔

#### 说明

使用固定循环时，一般操作顺序如下：

- 1. 工序  
X 轴, (Z 轴) 和 C 轴的定位
- 2. 工序  
快速运行至平面 R
- 3. 工序  
钻孔
- 4. 工序  
在钻孔底上进行加工

- 5. 工序

返回平面 R

- 6. 工序

快速返回定位平面

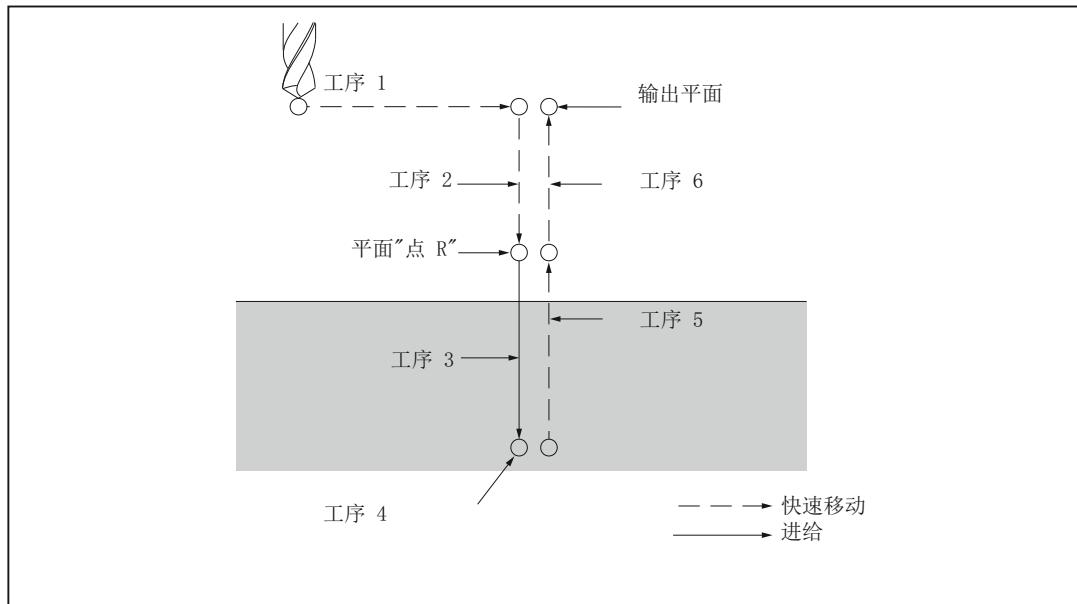


图 5-28 钻孔循环中的工序

#### 说明：定位轴和钻孔轴

如下所示，通过钻孔的 G 功能确定定位轴和钻孔轴。其中 C 轴和 X 轴或 Z 轴与定位轴相对应。钻孔轴由 X 轴或者 Z 轴构成：这些轴不能用作定位轴。

表格 5-4 相应钻孔轴的定位平面

G 功能	定位平面	钻削轴
G83, G84, G85	X 轴, C 轴	Z 轴
G87, G88, G89	Z 轴, C 轴	X 轴

G83 和 G87, G84 和 G88, 以及 G85 和 G89 的工序除了钻孔轴以外都相同。

## 其它功能

### 5.1 程序支持功能

#### 钻孔模式

G 功能 (G83-G85, G87-89) 一直保持模态有效，直到撤销对它们的选择。选择 G 功能时，钻孔模式一直保持有效。数据保留，直到改变或者取消钻孔循环中的钻孔数据。

必须在固定循环开始时设定所要求的所有钻孔数据。在执行固定循环的过程中，只能修改数据。

#### 重复

如果要对等间距的多个钻孔进行加工，可在参数“K”中设定重复次数。“K”只在定义该参数的程序段中生效。

存储钻孔数据：在编程 K0 时不执行钻孔。

#### 取消选择

通过 G80 或 G 功能组 01 的 G 功能 (G00, G01, G02, G03) 取消固定循环。

#### 符号和图像

下面对单个固定循环进行说明。这些符号会在下面的图像中使用：

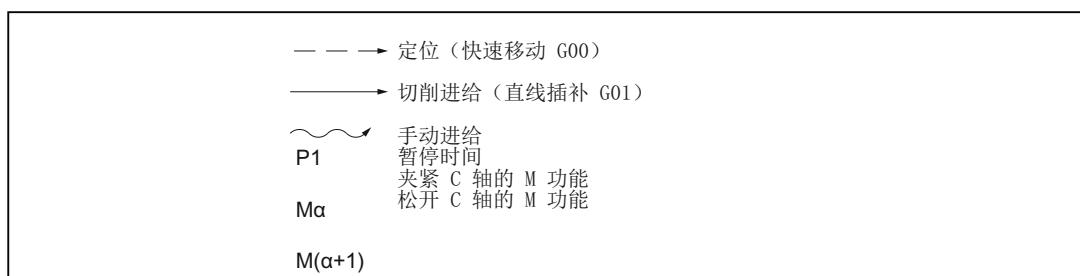
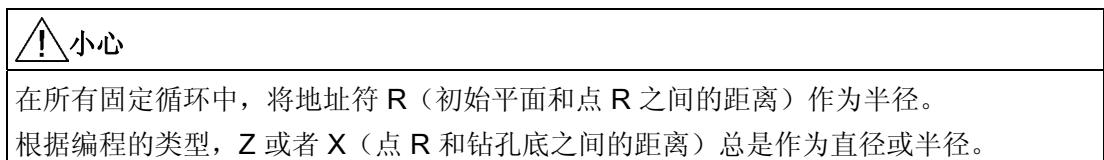


图 5-29 符号和图像



#### 深孔钻削循环 (G83) / 侧面深孔钻削循环 (G87)

是执行深孔钻削循环 (排屑) 还是高速深孔钻削循环 (断屑)，取决于 USER DATA, \_ZSFI[20] 的设置。

如果没有为钻孔循环设定进给率，执行标准钻孔循环。

### 高速深孔钻削循环 (G83, G87) (USER DATA, \_ZSFI[20]=0)

在高速深孔钻削循环中，钻头以切削进给率进行重复进给。钻头以固定量返回，直到刀具达到钻孔底。

#### 格式

G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... ;

或

G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... ;

**X, C 或 Z, C:** 孔的位置

**Z 或 X:** 点 R 到钻孔底的距离

**R\_:** 初始平面到平面 R 的距离

**Q\_:** 进给

**P\_:** 钻孔底的暂停时间

**F\_:** 切削进给率

**K\_:** 重复加工的次数 (需要使用时)

**M\_:** 夹紧 C 轴的 M 功能 (需要使用时)

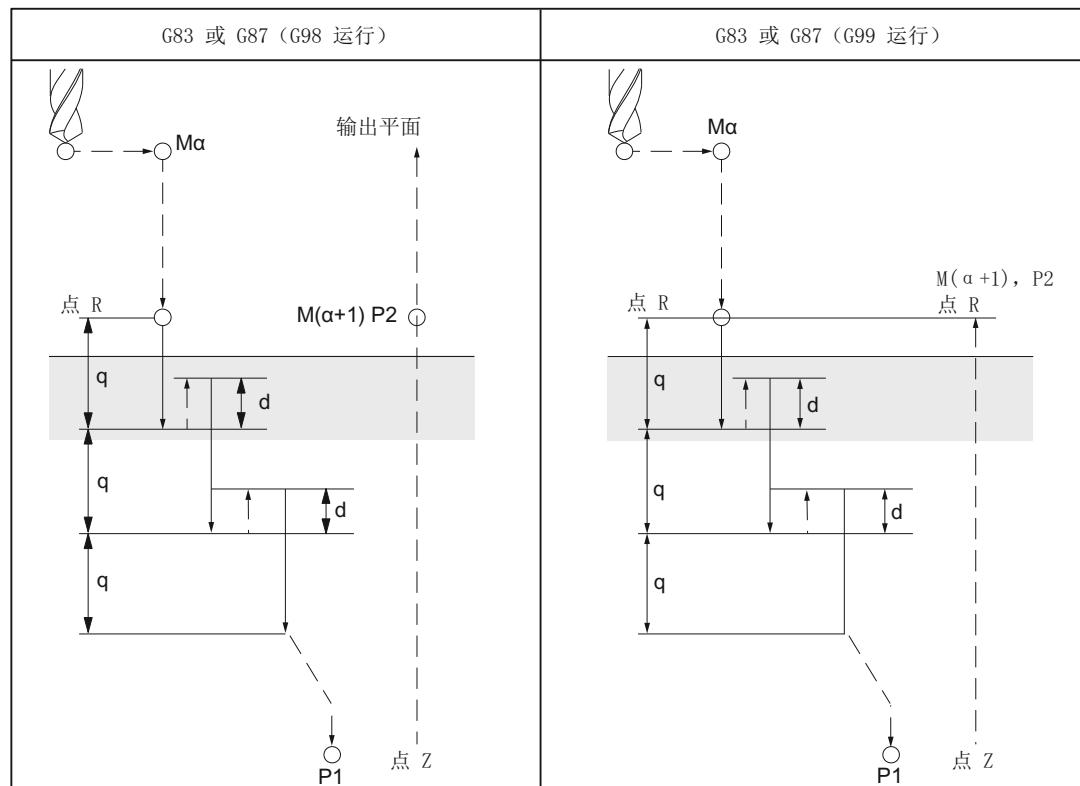


图 5-30 “高速深孔钻削”循环

---

## 5.1 程序支持功能

**M<sub>a</sub>:** 夹紧 C 轴的 M 功能

**M( $\alpha+1$ ):** 松开 C 轴的 M 功能

**P1:** 暂停时间 (程序)

**P2:** 在 USER DATA, \_ZSFR[22]中设定暂停时间

**d:** 在 USER DATA, \_ZSFR[21]中设定返回量

### 深孔钻削循环 (G83, G87) (USER DATA, \_ZSFI[20]=1)

在深孔钻削循环中, 钻头以切削进给率进行重复进给。钻头返回至平面 R, 直到刀具到达钻孔底。

#### 格式

G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... K... ;

或

G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... K... ;

**X, C 或 Z, C:** 孔的位置

**Z 或 X:** 点 R 到钻孔底的距离

**R\_:** 初始平面到平面 R 的距离

**Q\_:** 进给

**P\_:** 钻孔底的暂停时间

**F\_:** 切削进给率

**K\_:** 重复加工的次数 (需要使用时)

**M\_:** 夹紧 C 轴的 M 功能 (需要使用时)

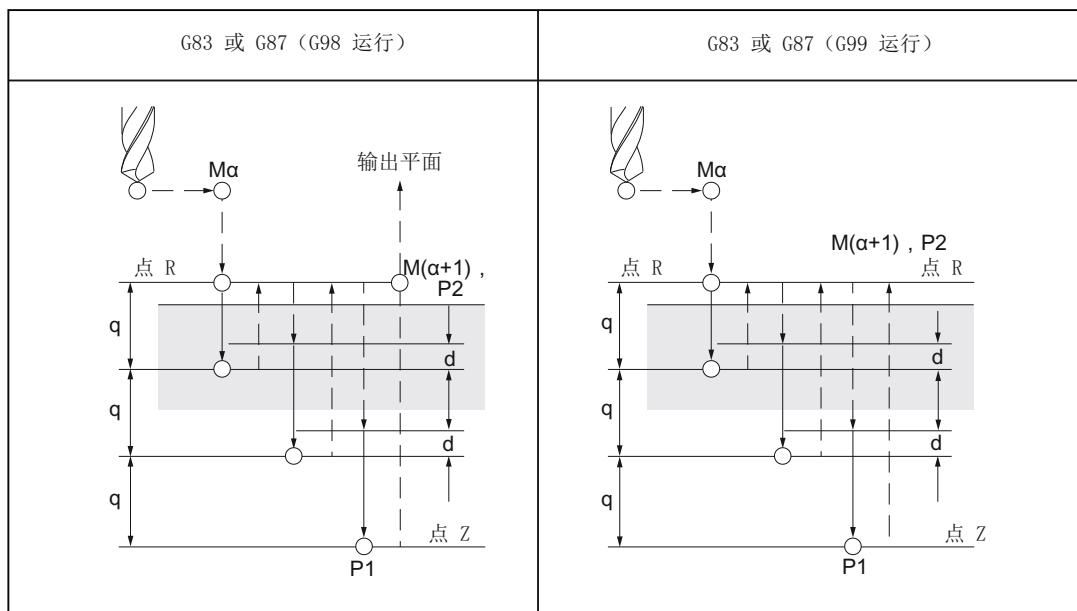


图 5-31 深孔钻削循环

**Mα:** 夹紧 C 轴的 M 功能**M( $\alpha+1$ ):** 松开 C 轴的 M 功能**P1:** 暂停时间 (程序)**P2:** 在 USER DATA, \_ZSFR[22]中设定暂停时间**d:** 在 USER DATA, \_ZSFR[21]中设定返回量

### 示例

```

M3 S2500 ; 钻头的旋转
G00 X100.0 C0.0 ; x 轴和 c 轴的定位
G83 Z-35.0 R-5.0 Q5000 F5.0 ; 孔 1 的加工
C90.0 ; 孔 2 的加工
C180.0 ; 孔 3 的加工
C270.0 ; 孔 4 的加工
G80 M05 ; 取消循环和
           ; 钻孔刀具停止
  
```

### 钻孔循环 (G83 或 G87)

如果没有为进给 (Q) 写入值，执行标准钻孔循环。在这种情况下，刀具从钻孔底快速返回。

## 其它功能

### 5.1 程序支持功能

#### 格式

G83 X(U)... C(H)... Z(W)... R... P... F... M... K... ;

或

G87 Z(W)... C(H)... X(U)... R... P... F... M... K... ;

**X, C 或 Z, C:** 孔的位置

**Z 或 X:** 点 R 到钻孔底的距离

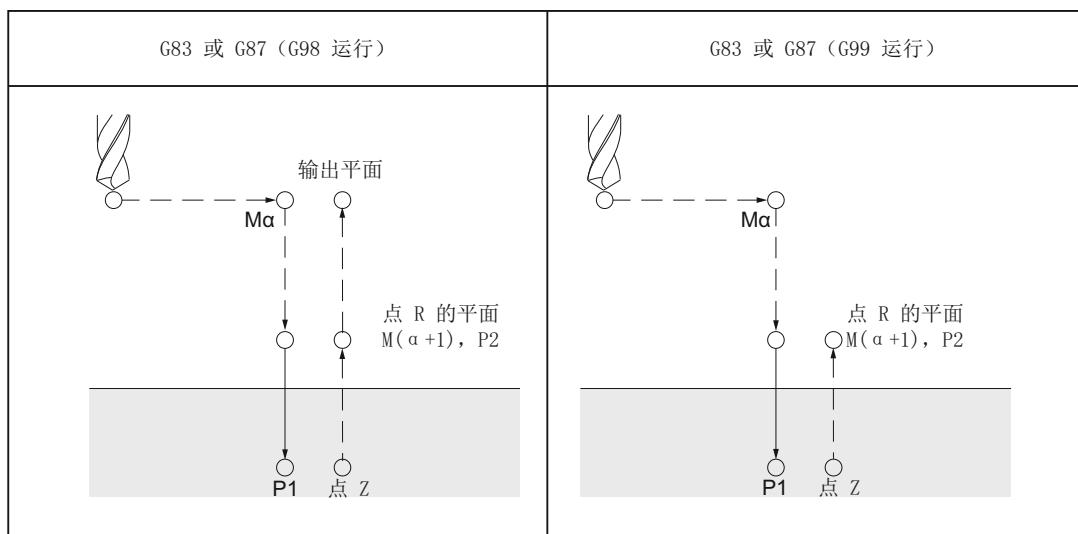
**R\_:** 初始平面到平面 R 的距离

**P\_:** 钻孔底的暂停时间

**F\_:** 切削进给率

**K\_:** 重复加工的次数（需要使用时）

**M\_:** 夹紧 C 轴的 M 功能（需要使用时）



**Ma:** 夹紧 C 轴的 M 功能

**M(a+1):** 松开 C 轴的 M 功能

**P1:** 暂停时间（程序）

**P2:** 在 USER DATA, \_ZSFR[22]中设定暂停时间

## 示例

M3 S2500	; 钻头的旋转
G00 X100.0 C0.0	; X 轴和 C 轴的定位
G83 Z-35.0 R-5.0 P500 F5.0	; 孔 1 的加工
C90.0	; 孔 2 的加工
C180.0	; 孔 3 的加工
C270.0	; 孔 4 的加工
G80 M05	; 取消循环和 ; 钻孔刀具停止

每次切削进给率 Q 都达到编程的切削深度后，刀具快速运行，返回参考平面 R。同样再次以快速运行进行下一次的切削进给，位移量(d)可以通过 USER DATA, \_ZSFR[10]给定。位移 d 和每次切削进给 Q 的切削深度都以切削进给率执行。设定 Q 为无符号的增量值。

### 说明

如果\_ZSFR[10]

- > 0: 使用间隔位移“d”（最小位移 0.001）
- = 0: 在循环内部对间隔位移 d 进行如下计算:
  - 钻孔深度为 30 毫米时，间隔位移的值始终为 0.6 毫米。
  - 对于更大的钻孔深度，使用公式钻孔深度/50 进行计算（最大值 7 毫米）。

## 螺纹钻孔循环端面 (G84), 侧面 (G88)

在该循环中，主轴旋转方向在钻孔底处反向。

### 格式

G84 X(U)... C(H)... Z(W)... R... P... F... M... K... ;

或

G88 Z(W)... C(H)... X(U)... R... P... F... M... K... ;

**X, C 或 Z, C:** 孔的位置

**Z 或 X:** 点 R 到钻孔底的距离

**R\_:** 初始平面到平面 R 的距离

**P\_:** 钻孔底的暂停时间

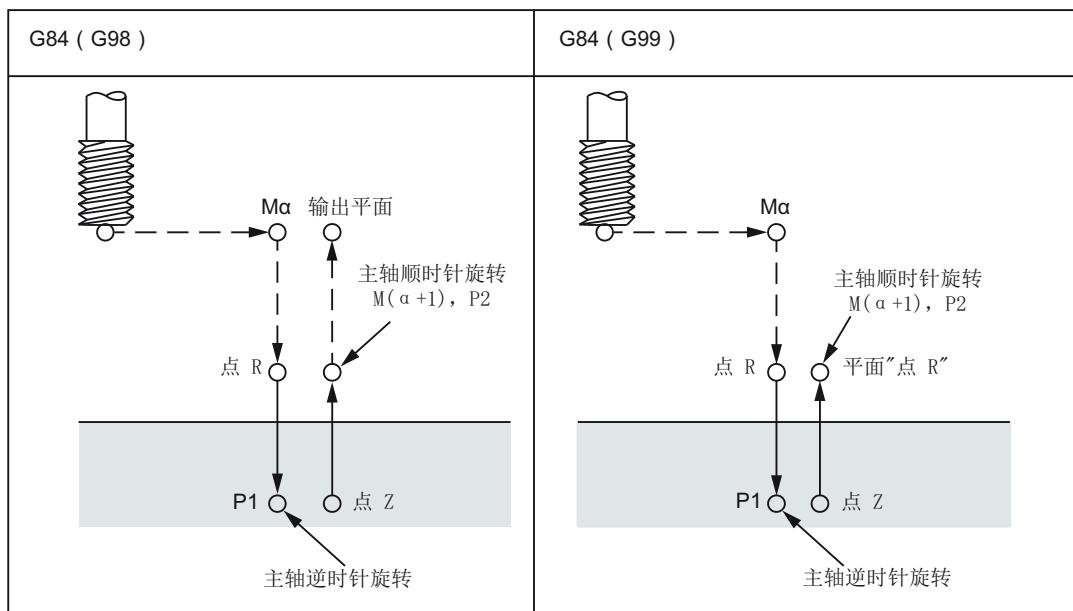
**F\_:** 切削进给率

**K\_:** 重复加工的次数（需要使用时）

**M\_:** 夹紧 C 轴的 M 功能（需要使用时）

## 其它功能

### 5.1 程序支持功能



P2: 在 USER DATA, \_ZSFR[22] 中设定暂停时间

### 说明

在螺纹钻孔中，主轴在钻孔底的方向上顺时针旋转；在返回时反向旋转。循环继续运行，直到刀具完全返回。

### 示例

M3 S2500	; 螺纹钻头的旋转
G00 X100.0 C0.0	; X 轴和 C 轴的定位
G84 Z-35.0 R-5.0 P500 F5.0	; 孔 1 的加工
C90.0	; 孔 2 的加工
C180.0	; 孔 3 的加工
C270.0	; 孔 4 的加工
G80 M05	; 取消循环和 ; 钻孔刀具停止

## 钻孔循环 端面 (G85), 侧面 (G89)

## 格式

G85 X(U)... C(H)... Z(W)... R... P... F... K... M... ;

或

G89 Z(W)... C(H)... X(U)... R... P... F... K... M... ;

**X, C 或 Z, C:** 孔的位置

**Z 或 X:** 点 R 到钻孔底的距离

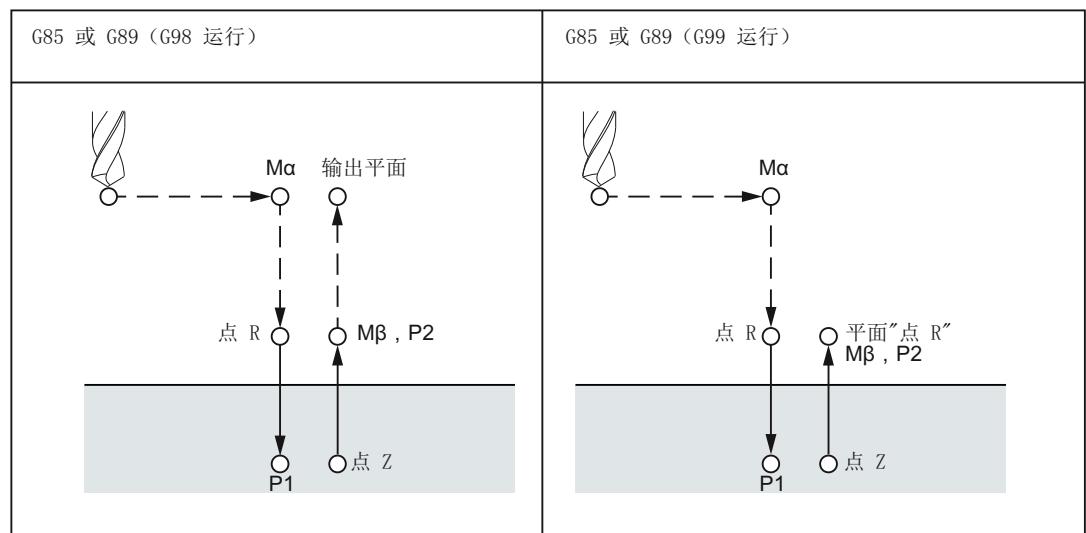
**R:** 初始平面到平面 R 的距离

**P:** 钻孔底的暂停时间

**F:** 切削进给率

**K\_:** 重复加工的次数 (需要使用时)

**M:** 夹紧 C 轴的 M 功能 (需要使用时)



P2: 在 USER DATA, \_ZSFR[22]中设定暂停时间

## 说明

在钻孔底上定位后, 向点 R 快速运行。接着从点 R 向点 Z 钻孔, 并返回点 R。

## 其它功能

### 5.1 程序支持功能

#### 示例

M3 S2500	; 钻头的旋转
G00 X50.0 C0.0	; X 轴和 C 轴的定位
G85 Z-40.0 R-5.0 P500 M31	; 孔 1 的加工
C90.0 M31	; 孔 2 的加工
C180.0 M31	; 孔 3 的加工
C270.0 M31	; 孔 4 的加工
G80 M05	; 取消循环和 ; 钻孔刀具停止

#### 取消钻孔固定循环 (G80)

通过 G80 取消钻孔固定循环。

#### 格式

G80;

#### 说明

钻孔固定循环取消，并重新过渡至标准运行。

## 5.2 可编程数据输入

### 5.2.1 更改刀具补偿值 (G10)

通过指令“G10 P...X(U)...Y(V)...Z(W)...R(C)...Q ;”可覆盖当前的刀具补偿。但是通过该指令不能创建新的刀具补偿。

表格 5-5 地址描述

地址	说明
P	刀具补偿号 (参见下面的说明)
X	X 轴的刀具补偿 (绝对, 增量)
Y	X 轴的刀具补偿 (绝对, 增量)
Z	Z 轴的刀具补偿 (绝对, 增量)
U	X 轴的刀具补偿 (增量)
V	X 轴的刀具补偿 (增量)
W	Z 轴的刀具补偿 (增量)
R	刀沿半径补偿 (绝对)
C	刀沿半径补偿 (增量)
Q	刀沿长度

#### 地址符 P

通过地址符 P 设定刀具补偿号，同时也设定，刀具几何数据或磨损量的补偿值能否更改。通过地址符 P 给定的值由 \$MC\_EXTERN\_FUNCTION\_MASK，位 1 中的设置决定：

\$MC\_EXTERN\_FUNCTION\_MASK, 位 1 = 0

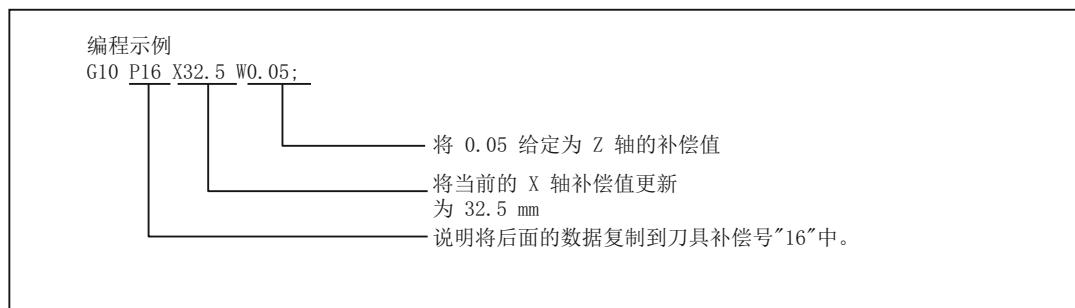
P1 至 P99: 写入刀具磨损量

P100 + (1 至 1500): 写入刀具磨损量

\$MC\_EXTERN\_FUNCTION\_MASK, 位 1 = 1

P1 至 P9999: 写入刀具磨损量

P10000 + (1 至 1500): 写入刀具磨损量



## 5.2 可编程数据输入

### 写入零点偏移

通过指令“G10 P00 X (U) ... Z (W) ... C (H) ... ;”可以在零件程序中写入和更新零点偏移。未编程轴的补偿值保持不变。

**X, Z, C:** 工件坐标系中的绝对或增量 (G91) 补偿值

**U, W, H:** 工件坐标系中的增量补偿值

### 5.2.2 用于调用子程序的 M 功能(M98, M99)

零件程序存储器中保存了子程序时，可以使用该功能。可以任意调用并执行存储器中保存的、属于相应程序号的子程序。

#### 指令

以下 M 功能可用于调用子程序。

表格 5- 6 用于调用子程序的 M 功能

M 功能	功能
M98	子程序调用
M99	结束子程序

#### 调用子程序(M98)

- M98 Pnnnnmmmm
  - m: 程序号（最多 4 位）
  - n: 重复执行的次数（最多 4 位）使用 M98 Pnnnnmmmm 调用程序前，须给程序正确命名，即，通常借用 0 将程序号补充为 4 位。
- 例如，如果写入了 M98 P21，则在零件程序存储器中查找程序名为 21.mpf 的程序，随后执行该子程序一次。如需执行该子程序三次，必须写入 M98 P30021。查找不到给定的程序号时，会输出一条报警信息。
- 允许子程序相互嵌套；最多可出现 16 个子程序级。设置的子程序级超出允许的最大值时，会输出一条报警信息。

#### 结束子程序(M99)

指令 M99 Pxeee 将结束子程序，并在已调用的程序中从程序段号 Nxeee 继续处理。数控系统首先向前（从调用子程序开始到结束程序）查找程序段号。如果没有查找到一致的程序段号，系统接着向后（零件程序开始的方向）查找零件程序。

如果主程序中的 M99 不带程序段号(Pxeee)，将从程序头跳转到主程序并重新开始主程序。主程序中的 M99 带程序段号跳转时(M99 Pxeee)，系统始终从程序头开始查找程序段号。

## 5.3 八位数的程序号

通过机床数据 20734 \$MC\_EXTERN\_FUNCTION\_MASK, 位 6=1 激活八位数的程序号选择。该功能作用于 M98、G65/66 和 M96。

y: 程序运行次数

x: 程序号

### 子程序调用

\$MC\_EXTERN\_FUNCTION\_MASK, 位 6 = 0

M98 Pyyyyyyyy 或者

M98 Pxxxxx Lyyyy

程序号最多为四位

通常借用 0 将程序号补充为 4 位

示例:

M98 P20012: 调用 0012.mpf, 运行 2 次

M98 P123 L2: 调用 0123.mpf, 运行 2 次

\$MC\_EXTERN\_FUNCTION\_MASK, 位 6 = 1

M98 PxXXXXXXXX Lyyyy

即使程序号的位数少于 4 位时，也不通过 0 进行补位。

不能在 P(Pyyyyyyyy) 中写入运行次数和程序号；运行次数必须通过 L 写！

示例:

M98 P123: 调用 123.mpf, 运行 1 次

M98 P20012: 调用 20012.mpf, 运行 1 次

**注意：**它和 ISO 原始编程指令不再匹配

M98 P12345 L2: 调用 12345.mpf, 运行 2 次

### 模态生效和程序段方式生效的宏 G65/G66

\$MC\_EXTERN\_FUNCTION\_MASK, 位 6 = 0

G65 Pxxxxx Lyyyy

即使程序号的位数少于 4 位时，也不通过 0 进行补位。程序号位数大于 4 位时输出报警。

\$MC\_EXTERN\_FUNCTION\_MASK, 位 6 = 1

G65 Pxxxxx Lyyyy

即使程序号的位数少于 4 位时，也不通过 0 进行补位。程序号位数大于 8 位时输出报警。

## 5.4 测量功能

### 中断 M96

\$MC\_EXTERN\_FUNCTION\_MASK,位 6 = 0

M96 Pxxxx

通常借用 0 将程序号补充为 4 位

\$MC\_EXTERN\_FUNCTION\_MASK,位 6 = 1

M96 Pxxxx

即使程序号的位数少于 4 位时，也不通过 0 进行补位。程序号位数大于 8 位时输出报警。

## 5.4 测量功能

### 5.4.1 通过 G10.6 快速退刀

通过 G10.6<轴位置>可以激活刀具快速退刀时（如出现断刀时）的退回位置。而退回运行由数字信号启动。NC 的第 2 快速输入用作启动信号。

也可通过机床数据 10820 \$MN\_EXTERN\_INTERRUPT\_NUM\_RETRAC (1 - 3) 选择其它快速输入(1 - 3)。

通过 G10.6 进行退刀时必须存在中断程序(ASUP) CYCLE3106.spf。如果在零件程序存储器中不存在程序 CYCLE3106.spf, G10.6 零件程序段会输出报警 14011“不存在程序 CYCLE3106 或没有释放处理”。

在 SUP CYCLE3106.spf 中可定义快速退刀后数控系统的特性。如果需要在快速退刀后停止轴和主轴，必须在 CYCLE3106.spf 中写入 M0 和 M5。如果 CYCLE3106.spf 是只包含 M17 的哑元程序，退刀后零件程序继续，并不中断。

如果通过写入 G10.6<轴位置> 激活了快速退刀，第 2 快速 NC 输入的输入信号从 0 变为 1 后当前运行被中断，以快速移动方式返回到 G10.6 写入的位置。此时，可按照绝对值方式或增量值方式运行到 G10.6 程序段中写入的位置。

通过 G10.6（无位置信息）可以取消该功能。通过第 2 快速 NC 输入的输入信号禁用快速退刀。

### 限制

只可以写入一个轴的快速退刀。

### 5.4.2 删 除剩余行程的测量 (G31)

通过“G31 X... Y... Z... F... ;”可执行带“删除剩余行程”的测量。若在线性插补中存在测量头 1 的测量输入，线性插补中断并且轴的剩余行程会被删除。程序从下一个程序段开始继续执行。

#### 格式

G31 X... Y... Z... F\_;

G31: 非模态生效 G 功能 (只在编程了该功能的程序段中生效)

#### PLC 信号“测量输入 = 1”

通过测量输入 1 的上升沿，将当前轴位置存储到轴系统参数或 \$AA\_MM[<Achse>]  
\$AA\_MW[<Achse>]中。可在西门子模式中读取该参数。

\$AA_MW[X]	存储工件坐标系中 X 轴的坐标值
\$AA_MW[Z]	存储工件坐标系中 Z 轴的坐标值
\$AA_MM[X]	存储机床坐标系中 X 轴的坐标值
\$AA_MM[Z]	存储机床坐标系中 Z 轴的坐标值

#### 说明

若在测量信号还有效的情况下激活了 G31，则会输出报警 21700。

#### 测量信号后的继续执行程序

如果在下一个程序段中写入了增量坐标轴位置，则该位置是以测量点为基准的。即，增量位置的参考点是测量信号执行“删除剩余行程”时所在的轴位置。

如果在下一个程序段中编程的是绝对轴位置，则轴运行到该位置。

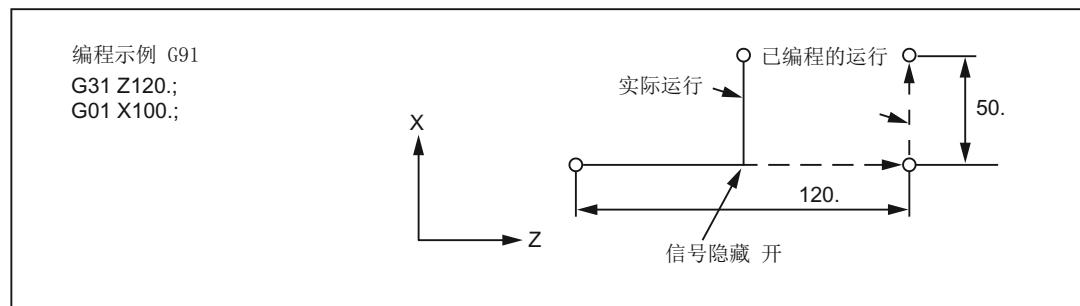


图 5-32 编程示例

---

## 5.4 测量功能

### 5.4.3 通过 G31, P1 - P4 测量

功能 G31 P1 (.. P4) 和 G31 唯一区别在于，它可以通过 P1 - P4 选择测量信号的不同输入。其中也可以同时监控多个输入上的测量信号上升沿。输入和地址 P1 - P4 之间的分配由机床数据确定。

#### 格式

G31 X... Y... Z... F... P... ;

X, Y, Z: 终点

F...: 进给率

P...: P1 - P4

#### 说明

通过机床数据将数字输入分配给地址 P1 - P4，如下：

P1: \$MN\_EXTERN\_MEAS\_G31\_P\_SIGNAL[0]

P2: \$MN\_EXTERN\_MEAS\_G31\_P\_SIGNAL[1]

P3: \$MN\_EXTERN\_MEAS\_G31\_P\_SIGNAL[2]

P4: \$MN\_EXTERN\_MEAS\_G31\_P\_SIGNAL[3]

该选择的详细说明（P1、P2、P3 或 P4）参见机床制造商的资料。

## 5.4.4 通过 M96/M97 (ASUP) 编程中断程序

### M96

通过 M96P<程序号>编程中断程序

通过外部信号启动该程序。在西门子模式提供的 8 个输入中，始终使用第 1 个快速 NC 输入来启动中断程序。通过机床数据 10818 \$MN\_EXTERN\_INTERRUPT\_NUM\_ASUP 也可选择其他快速输入（1-3）。

### 格式

M96 Pxxxx ; 激活程序中断

M97 ; 取消程序中断

这样在释放中断时首先调用 shell 循环 CYCLE396，此循环会调用在 ISO 模式中通过 Pxxxx 编程的中断程序。Shell 循环结束时，对机床数据 10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96,位 1 进行测定，并通过 REPOS 定位至中断点，或继续下一个程序段。

### M97

通过 M97 抑制中断程序的启动。只有在通过 M96 进行再次激活后，才能以外部信号启动中断程序。

如果要通过中断信号直接调用以 M96 Pxx 编程的中断程序（不执行 CYCLE396 的中间步骤），必须将机床数据 20734 \$MC\_EXTERN\_FUNCTION\_MASK 位 10 设为零。在西门子模式中，信号从 0 转换到 1 时，调用以 Pxx 编程的子程序。

通过机床数据设置中断功能的 M 功能号。通过机床数据 10804

\$MN\_EXTERN\_M\_NO\_SET\_INT 确定激活中断程序的 M 号，通过机床数据 10806

\$MN\_EXTERN\_M\_NO\_DISABLE\_INT 确定抑制中断程序的 M 号。

只可使用不是预留用于标准 M 功能的 M 功能号。M96 和 M97 是 M 功能的缺省设置。设置机床数据 10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96，位 0，用于激活功能。这样就不会向 PLC 输出 M 功能。如果未设置位 0，则 M 功能被视作普通的辅助功能。

中断程序结束后，一般情况下轴会运行至中断程序段后零件程序段的终点。如果要从中断点继续执行零件程序，必须在中断程序结束处写入一个 REPOS 指令，比如 REPOSA。为此必须在西门子模式下写入中断程序。

用于激活和取消中断程序的 M 功能必须单独写在程序段中。如果除了“M”和“P”外在程序段中还写了其他地址符，则会输出报警 12080（句法错误）。

## 5.4 测量功能

### 机床数据

通过下面的机床数据可确定中断程序功能的属性：

MD10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96:

- 位 0 = 0

不能执行中断程序，M96/M97 为普通的 M 功能。

- 位 0 = 1

允许激活 M96/M97 中断程序。

- 位 1 = 0

从中断程序段后的下一程序段终点位置继续执行零件程序（REPOSL RME）。

- 位 1 = 1

从中断位置继续执行零件程序。

(REPOSL RME)

- 位 2 = 0

中断信号立即中断当前程序段并启动中断程序。

- 位 2 = 1

在程序段结束时才启动中断程序。

- 位 3 = 0

出现中断信号时，加工循环立即中断。

- 位 3 = 1

加工循环结束时，才启动中断程序（在 shell 循环中测定）。

在 shell 循环中测定位 3，并对循环过程进行相应匹配。

在 shell 循环 CYCLE396 中测定位 1。

如果不通过 shell 循环 CYCLE396 调用中断程序（\$MC\_EXTERN\_FUNCTION\_MASK, 位 10 = 1），必须测定位 1；如果位 1 = TRUE，必须通过 REPOSL RMI 定位至中断点，否则必须通过 REPOSL RME 定位至程序段终点。

示例：

```
N100 M96 P1234      ;激活 ASUP 1234.spf。 快速输入
                      ;1. 的上升沿时
                      ;启动程序 1234.spf
"
"
N3000 M97          ;取消 ASUP
```

调用中断程序前不执行快速退刀（LIFTFAST）。根据 MD10808

\$MN\_EXTERN\_INTERRUPT\_BITS\_M96，通过中断信号的上升沿，立即启动中断程序。

## 限制

像普通子程序一样处理中断程序。也就是说，至少必须有一个可用的子程序级，用于执行中断程序。（提供 16 个程序级，以及 2 个预留用于 ASUP 中断程序的程序级）。

只有在中断信号沿从 0 变为 1 时，才启动中断程序。如果中断信号始终保持为 1，则不再重新启动中断程序。

## 5.5 宏程序

宏可以由多个零件程序段组成，并通过 M99 结束。通常宏是零件程序段中通过 G65 Pxx 或 G66 Pxx 调用的子程序。

通过 G65 调用的宏为程序段方式有效。通过 G66 调用的宏为模态有效，通过 G67 取消。

### 5.5.1 与子程序的区别

通过调用宏程序（G65, G66）可以给定可在宏程序中测定的参数。与此相反，调用子程序（M98）时不能设定参数。

### 5.5.2 调用宏程序(G65, G66, G67)

通常在调用宏程序后立即执行该程序。

调用宏程序的步骤参见下表。

表格 5-7 调用宏程序的格式

调用方法	指令代码	备注
单次调用	G65	
模态调用(a)	G66	由 G67 取消

#### 单次调用(G65):

##### 格式

G65 P\_ L\_ ;

通过给定“G65 P ... L... <依据>；”调用具备“P”程序号的宏程序，并执行“L”次。

必须在 G65 程序段中写入所需参数。

## 说明

在 G65 或 G66 零件程序段中，地址 Pxx 视为写入宏功能的子程序的程序号。通过地址 Lxx 可以定义宏调用的次数。该零件程序段的所有其它地址都视为过渡参数，其参数值保存在系统变量 \$C\_A 至 \$C\_Z 中。在子程序中可以读取这些系统参数并分析宏功能。如果在一个宏程序（子程序）中借助过渡参数调用了其它宏程序，则必须在调用新的宏程序前将子程序中的过渡参数保存在内部变量中。

调用宏时必须自动切换回西门子模式，从而可以使用内部的变量定义。在宏程序的第一行插入指令 PROC<程序名称>可以实现该要求。如果在子程序中写入了另一个宏调用，必须在此之前再次选择 ISO 编程指令模式。

表格 5-8 指令 P 和 L

地址	说明	数字的位数
P	程序号	4 位或者 8 位
L	重复次数	

## 地址 I、J、K 的系统参数

因为在一个包含宏调用的程序段中可以最多写入 10 次地址 I、J 和 K，因此必须借助数组索引查找到这些地址的系统变量。这三个系统变量的句法为 \$C\_I[.]、\$C\_J[.] 和 \$C\_K[.]。这些值在数组中按已编程的顺序排列。程序段中写入的地址 I、J 和 K 的数量分别保存在变量 \$C\_I\_NUM、\$C\_J\_NUM 和 \$C\_K\_NUM 中。

即使没有写入某个地址，用于调用宏的过渡参数 I、J、K 也被作为关联程序段处理。如果重新写入一个参数，或按照 I、J、K 的顺序写入下一个参数，则它属于下一个程序段。

为识别 ISO 模式中的程序顺序，应设置系统变量 \$C\_I\_ORDER、\$C\_J\_ORDER 和 \$C\_K\_ORDER。它们是和 \$C\_I、\$C\_K 相同的数组，并包含相应的参数号。

---

## 说明

只有在西门子模式的子程序中才可以读取过渡参数。

---

示例：

N5 I10 J10 K30 J22 K55 I44 K33

    Block1    Block2    Block3

    \$C\_I[0]=10

    \$C\_I[1]=44

    \$C\_I\_ORDER[0]=1

    \$C\_I\_ORDER[1]=3

    \$C\_J[0]=10

    \$C\_J[1]=22

    \$C\_J\_ORDER[0]=1

    \$C\_J\_ORDER[1]=2

    \$C\_K[0]=30

    \$C\_K[1]=55

    \$C\_K[2]=33

    \$C\_K\_ORDER[0]=1

    \$C\_K\_ORDER[1]=2

    \$C\_K\_ORDER[2]=3

---

#### 说明

\$C\_I[0]为 DIN 代码。要在 ISO 模式下使用该代码，必须设置机床数据 20734

\$MC\_EXTERN\_FUNCTION\_MASK, 位 3=1, 默认值为 800H。

---

### 循环参数 \$C\_x\_PROG

在 ISO 编程指令 0 模式中，取决于编程方式（整数或实数）可以按照不同方式计算写入的数值。通过机床数据激活不同的计算方法。

如果该 MD 置位，则数控系统的属性如下：

X100 ; X 轴运行 100 毫米（100 带小数点）=> 实数

Y200 ; Y 轴运行 0.2 毫米（200 不带小数点）=> 整数

如果程序段中写入的地址用作循环的过渡参数，则写入的值始终作为实数保存在 \$C\_x 变量中。输入整数时不再可以推断循环中的编程方式（实数或整数），因此也不会在通过换算系数计算写入的数值。

系统变量 \$C\_TYP\_PROG 可提供 REAL 或 INTEGER 编程方式的信息。

\$C\_TYP\_PROG 的结构完全和 \$C\_ALL\_PROG 以及 \$C\_INC\_PROG 一样。如果写入的值为 INTEGER，则该位变为 0。如果为 REAL，则变为 1。如果通过变量\$<编号>对数值进行编程，则相应位也变为 1。

#### 示例：

P1234 A100. X100 -> \$C\_TYP\_PROG == 1。

由于只写入了 A 的 REAL 值，因此只有位 0。

P1234 A100.C20. X100 -> \$C\_TYP\_PROG == 5。

位 1 和位 3 (A 和 C)

#### 限制：

在每个程序段中最多允许写入十个 I、J、K 参数。在变量 \$C\_TYP\_PROG 中，只为 I、J、K 分别配备一个位。因此，应在 \$C\_TYP\_PROG 中将 I、J、K 相应的位设置为 0。从而不推导 I、J 或 K 是 REAL 或 INTEGER。

### 模态调用(G66, G67)

通过 G66 可以调用一个模态生效的宏程序。只有满足规定的条件时，随后才执行给出的宏程序。

- 通过给定“G66 P... L... <参数>;”可以激活模态生效的宏程序。过渡参数的处理和 G65 相同。
- G66 由 G67 取消。

表格 5-9 模态调用的条件

调用条件	选择该运行的功能	取消该运行的功能
执行运行指令后	G66	G67

## 给定参数

写入地址 A - Z 可以确定过渡参数。

## 地址和系统变量之间的对应关系

表格 5-10 地址、变量、调用指令的地址间的对应关系

地址和变量之间的对应关系	
地址	系统变量
A	\$C_A
B	\$C_B
C	\$C_C
D	\$C_D
E	\$C_E
F	\$C_F
H	\$C_H
I	\$C_I[0]
J	\$C_J[0]
K	\$C_K[0]
M	\$C_M
Q	\$C_Q
R	\$C_R
S	\$C_S
T	\$C_T
U	\$C_U
V	\$C_V
W	\$C_W
X	\$C_X
Y	\$C_Y
Z	\$C_Z

### 地址和系统变量之间的对应关系

必须按照 I、J、K 的顺序才能使用 I、J 和 K。

由于在一个包含宏调用的程序段中可以最多给定 10 次地址 I、J 和 K，因此必须借助索引找到宏程序内该地址的系统变量。这三个系统变量的句法为 \$C\_I[.]、\$C\_J[.] 和 \$C\_K[.]。相应的数值按照写入顺序保存到表格中。程序段中写入的地址 I、J 和 K 的数量分别保存在变量 \$C\_I\_NUM、\$C\_J\_NUM 和 \$C\_K\_NUM 中。

和其它变量不同，在读取这三个变量时必须给出索引。循环调用（如 G81）通常使用索引 0。例如：N100 R10 = \$C\_I[0]

表格 5-11 地址、变量、调用指令的地址间的对应关系

地址和变量之间的对应关系	
地址	系统变量
A	\$C_A
B	\$C_B
C	\$C_C
I1	\$C_I[0]
J1	\$C_J[0]
K1	\$C_K[0]
I2	\$C_I[1]
J2	\$C_J[1]
K2	\$C_K[1]
I3	\$C_I[2]
J3	\$C_J[2]
K3	\$C_K[2]
I4	\$C_I[3]
J4	\$C_J[3]
K4	\$C_K[3]
I5	\$C_I[4]
J5	\$C_J[4]
K5	\$C_K[4]
I6	\$C_I[5]
J6	\$C_J[5]
K6	\$C_K[5]
I7	\$C_I[6]
J7	\$C_J[6]
K7	\$C_K[6]
I8	\$C_I[7]
J8	\$C_J[7]
K8	\$C_K[7]

地址和变量之间的对应关系	
I9	\$C_I[8]
J9	\$C_J[8]
K9	\$C_K[8]
I10	\$C_I[9]
J10	\$C_J[9]
K10	\$C_K[9]

### 说明

如果在多个程序段中写入地址 I、J、K，则每个程序段中 I/J/K 的顺序应使得变量号和该顺序相符。

### 示例：给定参数

不管何种地址，参数值也可以包含正负号和小数点。

参数值始终保存为实数值。

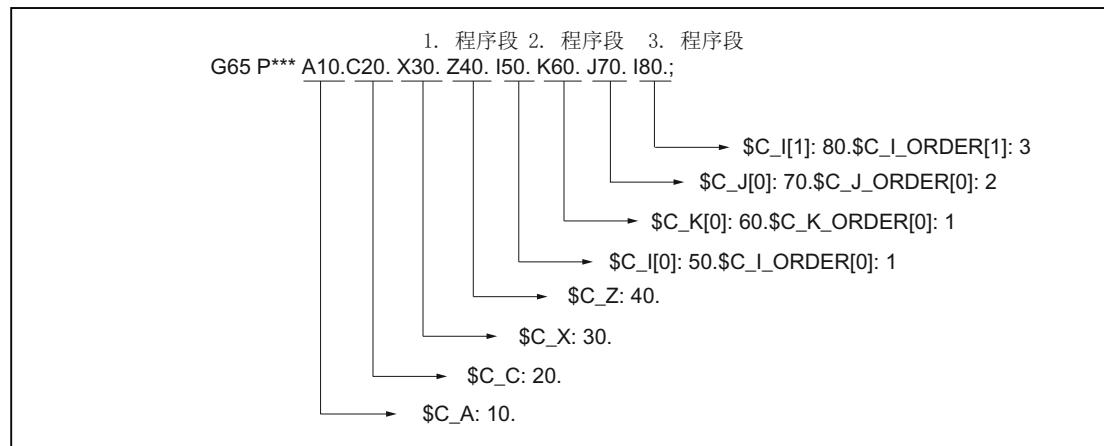


图 5-33     示例：给定依据

### 在西门子模式和 ISO 模式中执行宏程序

可以在西门子模式或 ISO 模式中调用已经被调用的宏程序。在宏程序的第一个程序段中确定执行程序的语言模式。

如果在宏程序的第一个程序段中包含指令 PROC <程序名称>，会自动切换入西门子模式。如果没有该指令，加工在 ISO 模式中进行。

在西门子模式中执行程序可以将过渡参数备份在本地变量中。而在 ISO 模式中却不可以将过渡参数保存在本地变量中。

如需读取在 ISO 模式中执行的宏程序的过渡参数，必须借助指令 G290 切换入西门子模式。

---

## 5.5 宏程序

### 举例

带有宏调用的主程序:

```
_N_M10_MP1:  
N10 M3 S1000 F1000  
N20 X100 Y50 Z33  
N30 G65 P10 F55 X150 Y100 S2000  
N40 X50  
N50 ....  
N200 M30
```

西门子模式中的宏程序:

```
_N_0010_SPF:  
PROC 0010 ; 切换入西门子模式  
N10 DEF REAL X_AXIS ,Y_AXIS, S_SPEED, FEED  
N15 X_AXIS = $C_X Y_AXIS = $C_Y S_SPEED = $C_S FEED = $C_F  
N20 G01 F=FEED G99 S=S_SPEED  
...  
N80 M17
```

ISO 模式中的宏程序:

```
_N_0010_SPF:  
G290; 切换入西门子模式  
; 读取过渡参数  
N15 X_AXIS = $C_X Y_AXIS = $C_Y S_SPEED = $C_S FEED = $C_F  
N20 G01 F=$C_F G99 S=$C_S  
N10 G1 X=$C_X Y=$C_Y  
G291; 切换入 ISO 模式  
N15 M3 G54 T1  
N20  
...  
N80 M99
```

## 5.6 附加功能

### 5.6.1 G05

与子程序调用“M98 Pxx”相似，通过指令 G05 可调用任意子程序。可对 G05 调用的子程序进行预编译，以加快程序的处理速度。

#### 格式

G05 Pxxxxx Lxxx ;

Pxxxxx：要调用程序的编号

Lxxx：重复次数

(如果未设定“Lxxx”，自动使用 L1。)

#### 示例

G05 P10123 L3 ;

通过此程序段调用程序 10123.mpf，并执行三次。

#### 限制

- 通过 G05 调用子程序时，在西门子模式下不发生转换。指令 G05 像通过“M98 P\_”调用子程序一样生效。
- 忽略包含 G05 而不含地址符 P 的程序段，不会输出报警。
- 同样忽略 G05.1 程序段（不管其中是否包含地址符 P）以及 G05 P0 程序段或 G05 P01 程序段，并且不输出报警。

## 5.6.2

## 多边形车削

使用多边形车削，可通过两条主轴的耦合来加工多边形工件。

通过编程句法 G51.2 Q.. P.. R.. 启动同步主轴耦合。通过参数 Q 和 P 确定从引导主轴到跟随主轴的传输比。如果以跟随主轴和引导主轴之间的偏差角启动耦合，需要通过地址 R 编程角度差。

在多边形车削中，边数不确定。比较典型的应用是四角螺栓头，或者六角螺栓或螺母。

通过编程 G51.2 总是将通道中的第 1 主轴定义为引导主轴，而将第 2 主轴定义为跟随主轴。选择额定值耦合作为耦合类型。

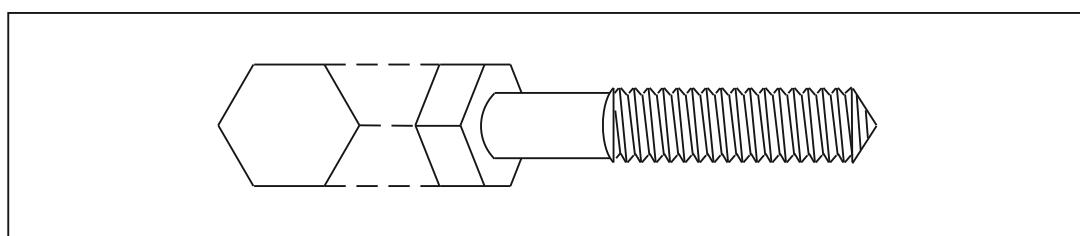


图 5-34 六角螺栓

## 格式

G51.2 P...Q...;

**P, Q:** 转速比例

通过地址符 Q 前的符号设定第 2 主轴的旋转方向。

## 示例

G00 X120.0 Z30.0 S1200.0 M03	; 将工件转速设置为 1200 转/分钟
G51.2 P1 Q2	; 刀具旋转开始 (2400 转/分钟)
G01 X80.0 F10.0	; X 轴进给
G04 X2.	;
G00 X120.0	; X 轴返回
G50.2	; 刀具旋转暂停
M05	; 主轴停止

不可在同一程序段中同时设定 G50.2 和 G51.2。

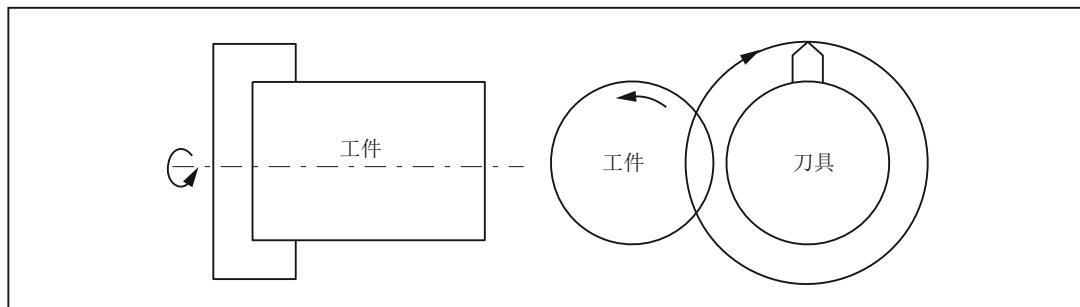


图 5-35 多边形车削

### 5.6.3 ISO 编程指令模式中的压缩程序

指令 COMPON、COMPCURV 和 COMPCAD 是西门子编程指令中的指令，它们可以激活压缩功能，将多个线性程序段综合成一个加工程序段。如果在西门子模式下激活了该功能，也可以在 ISO 编程指令模式下使用它来压缩线性程序段。

程序段最多由下列指令组成：

- 程序段号码
- G01, 模态或程序段方式生效
- 轴分配
- 进给率
- 注释

如果程序段中包含其它指令（例如，辅助功能或其它 G 代码等），不执行压缩功能。

允许通过 \$x 对 G、轴和进给率进行赋值，同样也适用于功能 Skip。

示例：压缩这些程序段

```

N5      G290
N10     COMPON
N15     G291
N20     G01 X100. Y100. F1000
N25     X100 Y100 F$3
N30     X$3 /1 Y100
N35     X100 (轴 1)

```

## 5.6 附加功能

不压缩这些程序段

```
N5      G290
N10    COMPON
N20    G291
N25    G01 X100 G17          ; G17
N30    X100 M22              ; 程序段中的辅助功能
N35    X100 S200             ; 程序段中的主轴转速
```

### 5.6.4 DryRun 和跳转级的切换模式

跳转级的切换(DB3200.DBB2)总是会影响程序运行，它会引起轨迹上短时间的速度骤降。同样 DryRun (DryRun = 空运行进给 DB3200.DBX0.6)的切换，即 DryRunOff 和 DryRunOn 之间的切换也会引起速度骤降。

现在使用一个具备有限功能的新切换模式可以避免出现该现象。

跳转级切换时，即 PLC 中的新值 -> NCK-Chan 接口信号 DB3200.DBB2，借助机床数据设置 10706 \$MN\_SLASH\_MASK==2 不再需要速度骤降。

---

#### 说明

NCK 分两个阶段处理程序段：预处理和主处理（也称为预运行和主运行）。预处理的结果进入预存处理器。主处理读取预处理器中最早的程序段并按照几何数据运行。

---

---

#### 说明

借助机床数据设置 \$MN\_SLASH\_MASK==2 可以在跳转级切换时切换预处理！按照旧的跳转级运行所有位于预处理器中的程序段。通常用户不能控制预处理器的数据容量。因此，用户会发现：切换结束后新的跳转级不知何时生效！

---

---

#### 说明

零件程序指令 STOPRE 可清空预处理器。如果在 STOPRE 前切换跳转级，STOPRE 的所有程序段肯定会被切换。隐含指令 STOPRE 也是如此。

---

借助机床数据 10704 \$MN\_DRYRUN\_MASK==2，切换 DryRun 模式时不需要速度骤降。但是此时也只切换预处理，并产生上文所提及的限制条件。因此同样：注意！

DryRun 模式切换后新的模式不知何时生效！

## 5.6.5 通过 M96, M97 写入中断程序

### M96

通过 M96 P<程序号>可以将一个子程序定义为一个中断程序。

通过外部信号启动该程序。在西门子模式中提供的八个输入中，中断程序始终使用第 1 个快速 NC 输入启动。通过 MD10818 \$MN\_EXTER\_INTERRUPT\_NUM\_ASUP 也可以选择其它的快速输入（1 至 8）。

### 格式

M96 Pxxxx	; 激活程序中断
M97	; 取消程序中断

M97 和 M96 P\_ 必须位于单独的程序段中。

从而可以在释放中断时首先调用 shell 循环 CYCLE396，而该循环随后调用 ISO 模式中写入的中断程序。Shell 循环结束后分析机床数据 10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96,位 1；以 REPOS 定位到中断点或继续下一个程序段。

### 结束中断(M97)

通过 M97 可以取消中断程序。只有再次由 M96 激活中断程序后中断程序才可由外部信号启动。

如果需要略去 CYCLE396 的中间步骤而直接用中断信号调用 M96 Pxx 写入的中断程序，必须设置机床数据 20734 \$MC\_EXTERN\_FUNCTION\_MASK,位 10。在西门子模式中，当信号从 0 -> 1 时调用 Pxx 写入的子程序。

中断功能的 M 功能号由机床数据设定。通过机床数据 10804

\$MN\_EXTERN\_M\_NO\_SET\_INT 确定激活中断程序的 M 号；通过机床数据 10806 \$MN\_EXTERN\_M\_NO\_DISABLE\_INT 确定抑制中断程序的 M 号。

只允许使用不是预留用于标准 M 功能的 M 号。M96 和 M97 是 M 功能的缺省设置。如需激活功能，必须置位机床数据 10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96 的位 0。M 功能随后不传送给 PLC。如果位 0 没有置位，M 功能被视为普通的辅助功能。

中断程序结束后，标准情况下轴会运行到中断程序段之后的零件程序段的终点。如果需要从中断点开始继续处理零件程序，必须在中断程序结束处写入一个 REPOS 指令，例如：REPOSA。为此必须在西门子模式中写入中断程序。

激活和取消中断程序的 M 功能必须位于单独的程序段中。如果除了 M 和 P 程序段中还写入了其它地址，会输出报警 12080（句法出错）。

## 5.6 附加功能

### 机床数据

通过以下机床数据可以确定中断程序功能的属性：

MD10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96:

位 0 = 0

| 不允许中断程序，M96/M97 是普通的 M 功能。

位 0 = 1

| 允许通过 M96/M97 激活中断程序。

位 1 = 0

| 零件程序从中断程序段后(REPOSL RME)的下一程序段终点继续执行。

位 1 = 1

| 零件程序从中断位置继续执行(REPOSL RMI)。

位 2 = 0

| 中断信号立即停止当前程序段并启动中断程序。

位 2 = 1

| 在程序段结束时才启动中断程序。

位 3 = 0

| 出现中断信号时立即停止加工循环。

位 3 = 1

| 在加工循环结束时才启动中断程序（在 shell 循环中分析）。

| 在 shell 循环中会分析位 3，并相应地匹配循环运行过程。

| 在 shell 循环 CYCLE396 中分析位 1。

如果中断程序不是由 shell 循环 CYCLE396 调用(\$MC\_EXTERN\_FUNCTION\_MASK, 位 10 = 1)，必须分析位 1。位 1 = TRUE 时，必须通过 REPOSL RMI 定位到中断点，否则应通过 REPOSL RME 定位到程序段终点。

示例：

N100 M96 P1234	; 激活 ASUP 1234spf。第 1 快速输入 ; 出现上升沿时， ; 启动程序 1234.spf
....	
....	
N300 M97	; 取消 ASUP

### 限制

如同普通子程序一样处理中断程序。也就是说，必须至少有一个可用的子程序级，才可以执行中断程序。（提供 16 个程序级以及两个预留用于 ASUP 中断程序的程序级。）

只有当中断信号从 0 变为 1 时，才启动中断程序。如果中断信号始终保持为 1，不再重新启动中断程序。

# 索引

## D

DryRun 模式, 122

G74, 86

G75, 88

G76, 89

## F

F 功能, 13

G80 到 G89, 92

G83, 94

G83 或 G87, 97

G83,G87, 95, 96

## G

G 代码系统 A, 9

G84, 99

G00, 13, 27, 29

G85, 101

线性插补, 29

G87, 94

G01, 29

G88, 99

G02, G03, 30

G89, 101

G04, 52

G92, 46

G05, 119

G92.1, 47

G10.6, 106

G94, 15

G20,G21, 51

G95, 15

G27, 37

G96,G97, 60

G28, 36

G97, 60

G30, 37

G 代码

G31, 107

显示, 8

G31, P1 - P4, 108

## I

G33, 38, 40, 41

ISO 语言模式, 7

G34, 43

## M

G40, G41/G42, 55

M 功能, 62

G53, 46

M00, 62

G65, G66, G67, 111

M01, 63

G70, 83

M02, 63

G71, 77

G72, 80

- M30, 63  
M96, 109  
M96, M97, 123  
M97, 109  
M98, M99, 104  
在一个程序段中给定多个 M 功能, 66  
多头螺纹切削, 41  
多次重复循环, 76  
多重螺纹切削循环, 89  
西门子模式, 7  
轨迹进给率, 13

## S

- Skip block (跳过程序段), 12  
S功能, 60

## X

- X 轴的直径和半径指令, 50

## 二划

- 刀具长度补偿, 54  
刀具补偿功能, 53  
刀具补偿数据存储器, 53  
刀沿半径补偿, 55

## 三划

- 子程序, 111  
小数点, 10

## 五划

- 主轴功能, 60  
功能程序中断, 123  
可变螺距的螺纹切削, 43  
用于停止操作的 M 功能, 62

## 六划

- 压缩功能, 121  
压缩器, 121

## 七划

- 坐标系, 45  
宏程序, 111  
宏程序调用, 111, 120  
快速退刀, 106  
快速移动, 13, 27  
每分钟线性进给率, 15  
附加功能, 62  
附加辅助功, 66

## 八划

- 侧面钻孔循环, 101  
侧面螺纹钻孔循环, 99  
其它 M 功能的用法, 66  
单次调用, 111  
参考点选择, 37  
图形重复循环, 86  
定位, 27  
注释, 12  
线性插补, 29  
英制/公制输入, 51

## 九划

- 恒定切削速度, 60  
轴运动的最大可编程数值, 9

## 十划

圆弧插补, 30  
通过 M96/M97 中断程序, 109

螺纹切削循环, 70, 77, 82

横向轴, 80

## 十一划

基本坐标系, 46  
控制点, 56  
旋转进给率, 15  
检查回参考点, 37

## 十二划

插补指令, 27  
暂停时间, 52  
程序段跳转级, 12

## 十三划

跳转级, 122

## 十四划

模态调用, 114  
端面钻孔循环, 101  
端面螺纹钻孔循环, 99  
精加工循环, 83

## 十五划

横向车削循环, 74  
横向轴上的深孔钻削和冲孔, 88  
横向圆锥车削循环, 75

## 十七划

螺纹切削, 38  
螺纹切削功能, 38

