

目录

1 概要.....	3
1.1 字符集.....	3
1.2 地址.....	3
1.3 字.....	3
1.4 代码与数据.....	3
1.5 关于坐标系.....	4
1.6 关于注释.....	4
2 段.....	5
2.1 定义.....	5
2.2 一个段内的约束.....	5
3 顺序号.....	6
4 段跳过指令“/”.....	7
5 G 代码.....	8
5.1 G00（定位、移动轴）	9
5.2 G01（直线插补）	10
5.3 G02, G03(圆弧插补指令)	10
5.4 G05、G06、G08、G09(轴镜像、X-Y交换, 取消镜像和轴交换)	11
5.5 G11,G12 (跳段)	13
5.5.1 G20, G21(单位选择).....	13
5.5.2 G28,G29(尖角过渡策略).....	13
5.5.3 G30,G31(延长指定距离).....	13
5.5.4 G34,G35 (减速加工的开始和取消)	13
5.5.5 G36,G37 (在拐点延长给定距离)	14
5.6 G40,G41,G42 (补偿和撤消补偿).....	16
5.7 G50, G51, G52（锥度加工）.. ..	20
5.8 G54 G55 G56 G57 G58 G59 (工作坐标系0~5).....	21
5.9 G60 G61(上下异形)	22
5.9.1 G74, G75 (四轴联动)	23
5.10 G90(绝对坐标指令)、G91 (增量坐标指令)	24
5.11 G92(设置当前点的坐标值).....	25
6 X、Y、U、V (I、J) 坐标轴.....	25
7 锥度加工.....	26
7.1 锥度加工实例.....	26
7.2 锥度加工的开始.....	26
7.3 锥度加工的连接.....	28
7.4 锥度和转角R	29
7.5 恒锥度和变锥度.....	31

8 M 代码	31
8.1 M00 (暂停指令)	31
8.2 M02(程序结束)	31
8.3 M98(子程序调用)	32
8.4 M99(子程序结束)	32
8.5 M50(剪丝代码)	32
8.6 M60(穿丝代码)	32
9 C 代码	32
10 子程序	33
11 关于运算	34
11.1 运算符地址	34
11.2 优先级	34
11.3 运算式的书写	34
12 H 代码(补偿)	34
13 代码的初始设置	35
14 转角功能	36
15 T 代码	37

1 概要

1.1 字符集

本系统编程中能够使用的字符如下：

数字字符：0 1 2 3 4 5 6 7 8 9

字母字符：A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

特殊字符：+ - ; / 空格 . ()

注意：本系统编程中，小写英文字母与大写英文字母所表示的意义相同。

1.2 地址

地址，是由一个或多个字母(A～Z)组成，用于定义附在其后的数据或代码的意义。本系统中可用的地址和意义，如下表：

地 址	意 义	地 址	意 义
N	顺序号	A	指定加工锥度
G	准备功能	R	转角 R 功能
X, Y, Z, U, V	表示轴	M	辅助功能
I, J	指定圆弧中心坐标	C	指定加工条件
H	偏移量指定	L	指定子程序调用的次数
P	指定子程序调用		

1.3 字

所谓字，就是一个地址后接一个相应的数据的组合体，它是组成程序的最基本单位。

字＝地址＋数据

例如：G00，M02，G01，X17.88 等。

1.4 代码与数据

代码和数据的输入形式如下：

C：表示加工条件号，其后接三位十进制数。从C001～C100 共100 种。

例如：C001，C096

H：指定偏移量值代码，其后可接三位十进制数，每一个变量代表一个具体的值，共有

H000～H099 共100 种。例如：H000，H009

G：准备功能，其后接二位十进制数，可表示直线或圆弧插补。例如：G00，G01，G02，等。

L：子程序重复执行次数，其后可接1～3 位十进制数，最多可调用999 次。例如：L5，L999。

M：辅助功能代码，其后接二位十进制数，如M00，M02 等。

N：程序的顺序号，其后接四位十进制数。例如：N0000，N1000 等。

P：指定调用子程序的序号，其后接四位十进制数。例如：P0001，P0100

X, Y, Z, U, V：坐标值代码，用以指定坐标移动的数据，其后接的数据在±999.999mm 或±99.9999 英寸范围内。

I,J: 表示圆弧中心坐标，其后数据可以在±99999.999mm 或±9999.9999 英寸之间。

例如：I5, J10。

A: 指定加工锥度，可输入0。

R: 转角 R 功能，后接的数据为所插圆弧的半径，最大为 999.999mm。

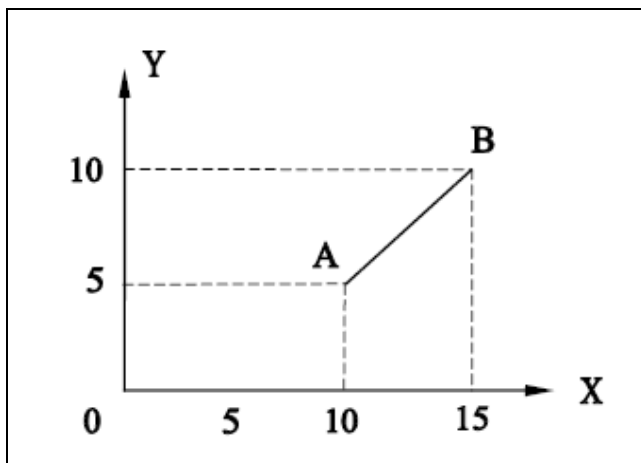
1.5 关于坐标系

本系统中有二种坐标系，绝对坐标系和增量坐标系。

所谓绝对坐标系，即每一点的坐标值都是以所选坐标系的原点为参考点而得出的值。

所谓增量坐标系，则是指当前点的坐标值是以上一个点为参考点而得出的值。

两种方式可用右图表示：



由A 点运动到B 点，在不同方式下程序如下：

- 绝对方式：G90 G01 X15. Y10.;
- 增量方式：G91 G01 X5. Y5.;

1.6 关于注释

在“(”和”)”之间的字符，为NC 程序的注释部分，它不是执行对象，仅表示对该NC 程序的注释和说明

```
列：（M02 STOP）.....注释
T84T86;
G90G92X0Y0;(1)..... 注释
M98P0002;
M00;
G00X50. ;(2)..... 注释
M00;
M98P0002;
M00;
G00Y-50. ;(3).....注释
M00;
```

```

M98P0002;
M00;
G00X0; (4) ..... .. 注释
M00;
M98P0002;
T85T87;
M02;

N0001; (子程序) ..... .. 注释
T84T86;
G90G92X0Y0;
C004;
G01X5. Y5. ;
      X5. Y-5. ;
      X-5. Y-5. ;
      X-5. Y5. ;
      X5. Y5. ;
      X0Y0;
T85T87;
M99;

N0002; (子程序) ..... .. 注释
T84T86;
G90G92X0Y0;
C004;
G01X5. ;
G02X-5. Y0I-5. J0;
      X5. Y0I5. J0;
G01X0;
T85T87;
M99;

```

2 段

2.1 定义

段，就是由一个地址或符号“/”开始，以“;”结束的一行程序。一个NC 程序由若干个段组合而成。

2.2 一个段内的约束

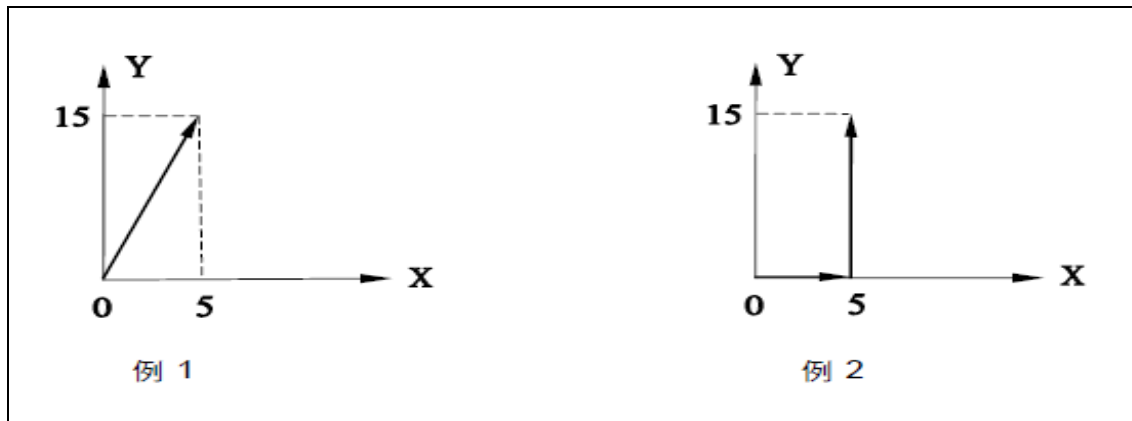
若在一段内含有 X，Y，U，V 轴中的任意两个或多个，依据代码，可多轴同时处理。

例1: G91 G00 X5. Y15.(表示X 轴, Y 轴同时移动5 mm, 15 mm), 若要按X、Y 轴的顺序移动, 请放在不同的段中。

例 2: G91 G00 X5.;

Y15.;

二者的运动轨迹不同, 请看下图所示:



[2] 在一个段内不能有二个运动代码, 否则将会出错。

例: G00 X10. G01 Y-10.; 则出错

应为: G00 X10.;

G01 Y-10.;

[3] 在同一个段中若有相同的轴标志, 则会出错。

例: G01 X10. Y20. X40.; 出错

3 顺序号

顺序号, 就是加在每个程序段前的编号, 可以省略。顺序号用N 开头, 后接四位十进制数, 以表示各段程序的相对位置, 这对查询一个特定程序很方便, 使用顺序号有以下两种目的:

[1] 用作程序执行过程中的编号;

[2] 用作调用子程序时的标记编号。

例: N0000;

G90 G92 X0. Y0. ;

C001;

M98 P0010;

C010;

M98 P0020;

M02;

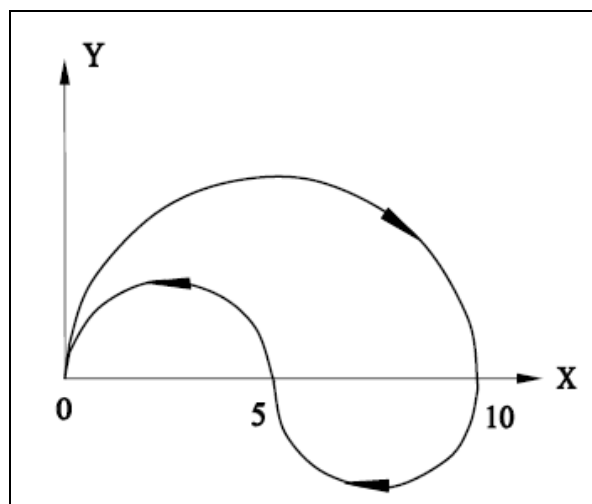
;

N0010;.....子程序标记号

G01 X0;

M99;

;



```

N0020;
G02 X10. I5.;
G02 X5. I-2.5;
G03 X0 I-2.5;
M99;

```

4 段跳过指令“/”

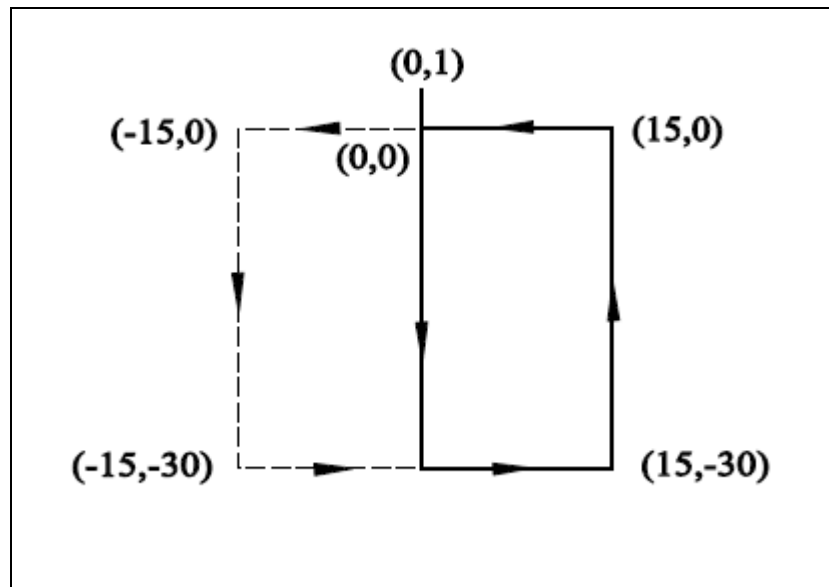
在程序中使用了 G11 代码时，在段首置有“/”标志的程序段将不执行，即自动跳过该程序段。程序中如未使用 G11 代码或使用了 G12，则执行此段程序。

注意：“/”只能位于一个段的段首。

```

G90 G92 X0 Y1.;
G11;
M98 P0030;
M02;
;
N0030;
G01 Y0;
/X-15.;
Y-30.;
X15.;
Y0;
X0;
Y1.;
M99;

```



如主程序中没有 G11（或者使用了 G12）代码，则子程序加工路径的左半部分应如虚线所示。

5 G 代码

G 代码，大体上可分为两种类型：

[1] 只对指令所在程序段起作用，称为非模态。

[2] 直到同一组中其它G 代码出现前，这个G 代码一直有效，称为模态。

例如：

G00 X10.;
Y10.;

G00 一直有效

G01 X20.;

G01 有效;

详细情况见下面的 G 代码一览表。

G 代码一览表

G 代码	功能	属性
G00	快速移动，定位指令	模态
G01	直线插补	模态
G02	顺时针圆弧插补指令	模态
G03	逆时针圆弧插补指令	模态
G04	暂停指令	
G05	X 镜像	模态
G06	Y 镜像	模态
G08	X--Y 交换	模态
G09	取消镜像和 X--Y 交换	模态
G11	SKIP ON (打开跳转)	模态
G12	SKIP OFF (关闭跳转)	模态
G20	英制	模态
G21	公制	模态
G28	尖角圆弧过渡	模态
G29	尖角直线过渡	模态
G30	取消 G31	模态
G31	延长给定距离	模态
G34	开始减速加工	模态
G35	取消减速加工	模态
G36	取消 G37	模态
G37	在拐角点延长给定距离	模态
G40	取消补偿	模态
G41	电极左补偿	模态
G42	电极右补偿	模态
G50	取消锥度	模态
G51	左锥度	模态
G52	右锥度	模态
G54	选择工作坐标系 1	模态
G55	选择工作坐标系 2	模态

G56	选择工作坐标系 3	模态
G57	选择工作坐标系 4	模态
G58	选择工作坐标系 5	模态
G59	选择工作坐标系 6	模态
G60	上下异形 OFF	模态
G61	上下异形 ON	模态
G74	四轴联动打开	模态
G75	四轴联动关闭	模态
G90	绝对坐标指令	模态
G91	增量坐标指令	模态
G92	指定坐标原点	

5.1 G00（定位、移动轴）

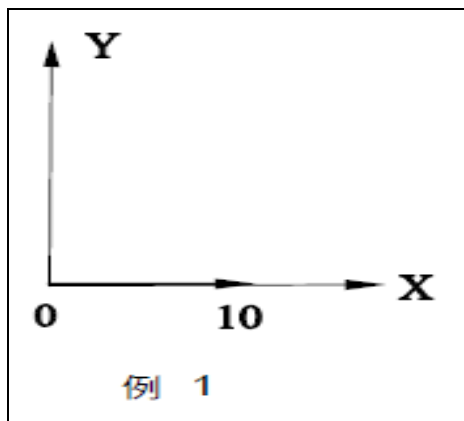
G00 代码为定位指令，用来快速移动轴。执行此指令后，不加工而移动轴到指定的位置。

指令格式：N**** G00{轴1}±{数据1}+{轴2}±{数据2}+.....

最多支持四个轴联动，即 X，Y，U，V 轴联动。

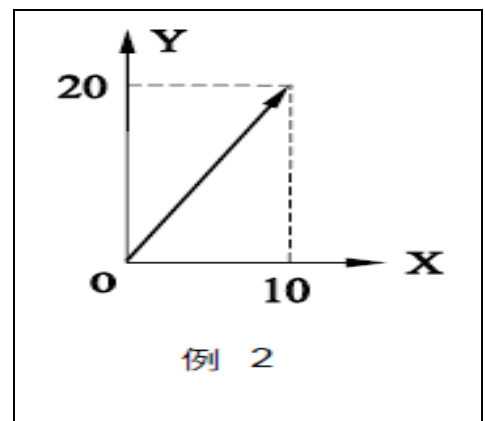
例 1： G00 X10.;

（一个轴移动）



例 2： G00 X10. Y20.;

（两个轴移动）



注意：在程序中，每个轴标志和它后面的数据之间不能有空格或其它字符，否则认为出错，若数据为正时，“+”号可以省略。

例3：出错情形：G00 X10.0 X20.0 ;

同一个段中有两个X 轴标志

例4：出错情形：G00X 10.;

出错，轴标志和数据之间有空格。

例5：出错情形：G00 XA10.;

出错，轴标志和数据之间出现别的字符。

5.2 G01（直线插补）

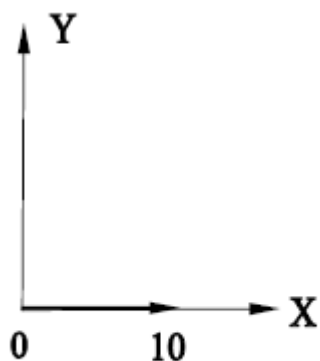
用G01 代码，可指令各轴直线插补加工。

格式：G01{轴}±{数据}

其后最多可以有四个轴标志和四个数据，可进行单轴、双轴及四轴直线插补加工。

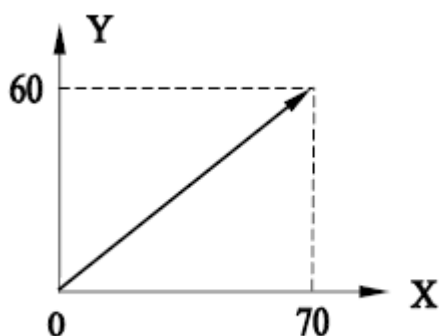
例1） 单轴加工 C001；

G01 X10.;



例2） 双轴加工 C001;

G01 X70. Y60.;



5.3 G02, G03(圆弧插补指令)

G02, G03 代码，用于指令圆弧插补加工。

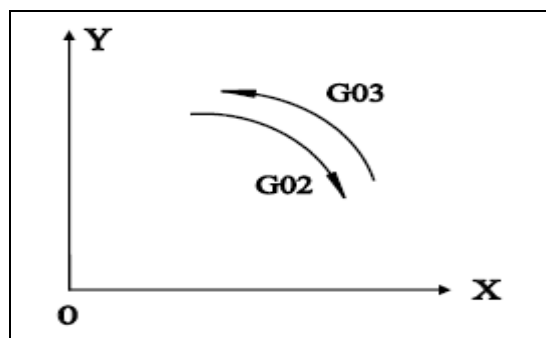
编程格式：{圆弧方向}{ 终点坐标}{圆弧圆心相对于圆弧起点的增量型坐标}

G02/G03 X - Y - I - J

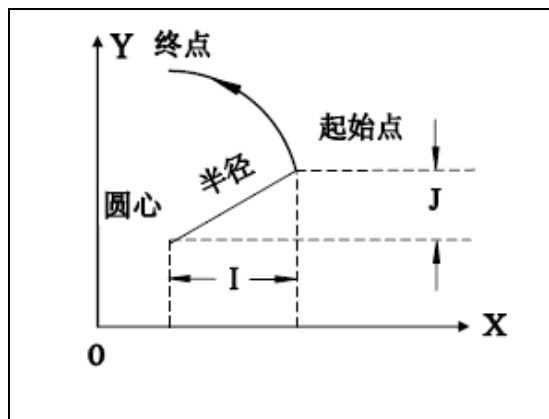
G02: 表示顺时针方向加工

G03: 表示逆时针方向加工

如右图表示：



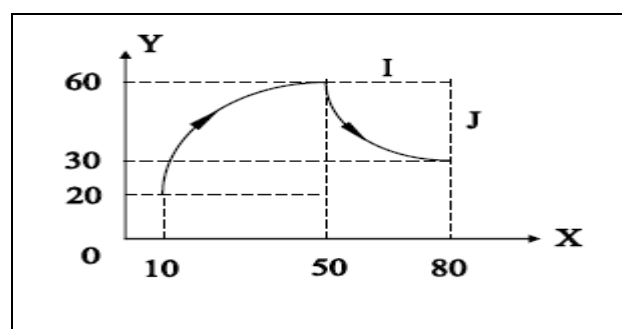
终点位置用X、Y 坐标表示，对应于G90，G91 分别用绝对、增量来表示，在用增量坐标时(G91)，终点坐标是相对于圆弧起点的坐标值。



圆心坐标相对于X、Y 轴分别用I、J 表示，它是一组相对于圆弧起点的增量型坐标值，如右图表示：

例：如图所示的轨迹，其程序为：

```
G90 G92 X10. Y20.;
G02 X50.0 Y60.0 I40.0
G03 X80.0 Y30.0 I30.0
```



注意：当 I, J 中有一个为零时，可以省略不写。

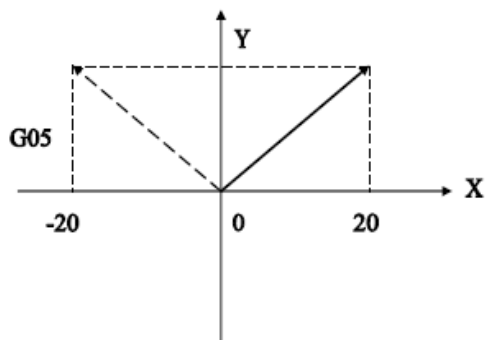
5.4 G05、G06、G08、G09(轴镜像、X-Y 交换，取消镜像和轴交换)

G05, G06 为图形的X、Y 轴镜像，G08 为图形关于X、Y 轴交换，G09 为取消镜像和轴交换，这组代码只对G00、G01、G02、G03 有影响。

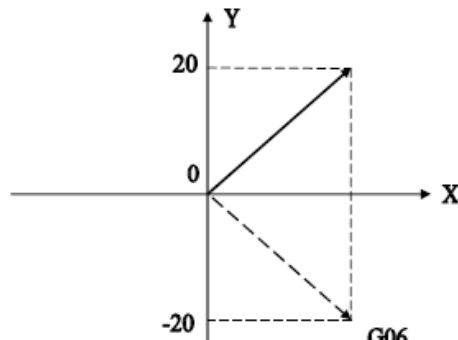
G05: 图形X 轴镜像即实际图形为程序轨迹关于Y 轴的对称图形。也就是说在实际走的轨迹中，X 值为程序中X 值的相反数。

G06: 图形Y 轴镜像，即实际图形为程序轨迹关于X 轴的对称图形。

例：直线插补的镜像图形：

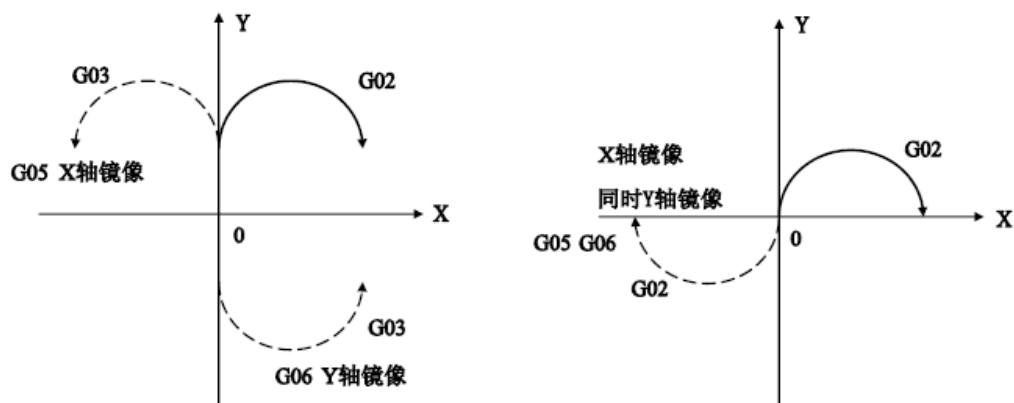


X轴镜像

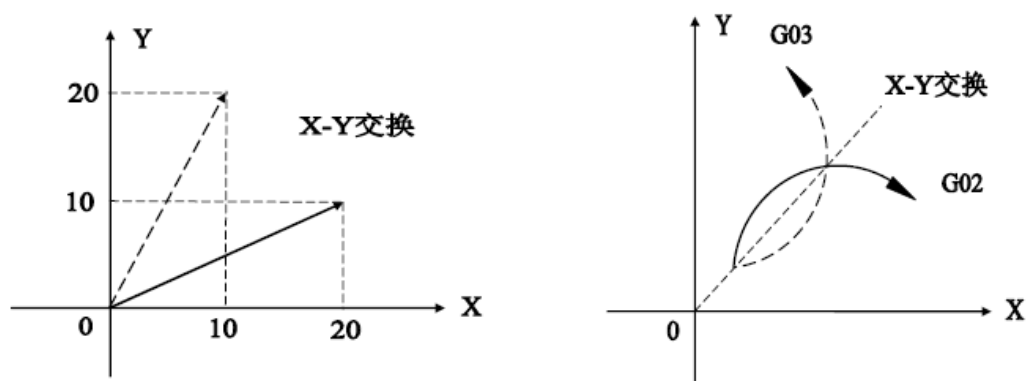


Y轴镜像

圆弧插补的镜像图形：



G08: 图形X、Y 轴交换，即程序中的X 轴实际走Y 轴地址所赋的值，程序中的Y 轴实际走X 轴地址所赋的值。



注意：

[1] 对圆弧插补来说，X 镜像和Y 镜像后G02 变成了G03，G03 变成了G02，X、Y 同时镜像后，则不变。

[2] X 轴和Y 轴同时镜像，请使用G05，G06 代码，即X 轴镜像后再Y 轴镜像，也可用G06，G05 代码，Y 轴镜像后再X 轴镜像，二者结果相同。

[3] 取消图形镜像和X-Y 交换请用G09 代码。

5.5 G11, G12 (跳段)

G11, G12 决定对段首有“/”的程序段是否忽略，即跳过。当用G11 代码时，表示要跳过段首有“/”的程序段，而不去执行该段程序；用G12 代码时，表示忽略段首的“/”符号，即段首有“/”的程序段照常执行，就像没有“/”一样。

注：执行G11 后，标志参数画面的SKIP 变成ON。执行G12 后，标志参数画面的SKIP 变成OFF。

例：G90 G92 X0 Y0;

G11;

/G41 H001;.....(A)

G01 X10.;

Y10.;

/ G02 X20. Y20. I10.;..... (B)

/M00;.....(C)

G01 Y30.;

X0;

G40 G01 Y0;

M02;

上述程序中(A)、(B)、(C)程序段将不执行。

5.5.1 G20, G21(单位选择)

G20, G21 这组代码决定在输入移动或加工尺寸时，是以英寸还是以公制(毫米)为单位。

G20	以英寸为单位
G21	以毫米为单位

注：[1] 这组代码请放在NC 程序的开头：

[2] 在以毫米为单位时，1 mm 可以写成1.，或1000、1.0。

在以英寸为单位时，1 英寸写成1.，或10000、1.0。

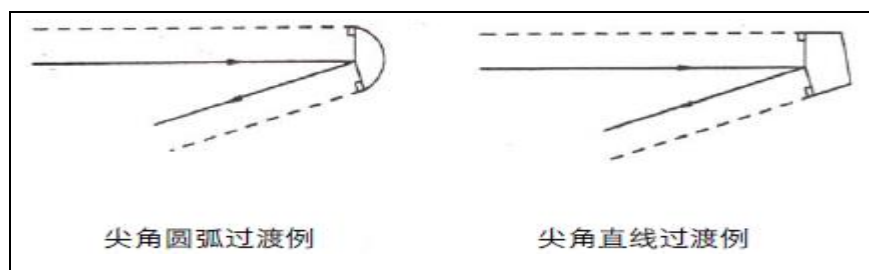
[3] 1 英寸=25.4 mm

5.5.2 G28,G29(尖角过渡策略)

G28, G29 代码用来选择尖角处理时的过渡策略。

G28	尖角圆弧过渡
G29	尖角直线过渡

尖角圆弧过渡即在尖角处加一个过渡圆，尖角直线过渡即在尖角处加三段直线，以确保尖角不受损伤。圆弧过渡和直线过渡的例子如下：



注：[1] 尖角过渡缺省为圆弧过渡，即一开机后自动设为圆弧过渡。

[2] 当补偿值为0 时，尖角过渡无效，即无尖角过渡。

5.5.3 G30,G31(延长指定距离)

G31	在G01 的直线段的终点按该直线方向延长给定距离
G30	取消G31

格式：G31X___

X 后的数为要延长之距离，该距离为大于或等于零之数，与坐标数据一致。

如：G31X30；表示要延长30 μ m。

G31 应放在要延长的直线段的开始。

如：G01X0；

G41H000；

G31X1.0X10.0Y10.0；□从本段开始，每段线段的终点延长1.0mm。

.....

G30G01X0；□从本段开始，取消延长。

当在NC 程序的开始增加：G31X0；则对于直线及圆弧均不进行内角/外角的特殊处理。如NC 程序中无G31X___，则自动进行内角/外角的缺省处理，这样所加工工件会有较明显的痕迹，但不影响尺寸精度及粗糙度，其尖角比较好。在NC 程序中如无G31，则用缺省处理方式。

G30：取消G31 功能。

5.5.4 G34,G35 (减速加工的开始和取消)

G34：自G01/G02/G03 的结束前3mm 处开始减速加工直到该段结束。

G35：取消G34 的减速加工。

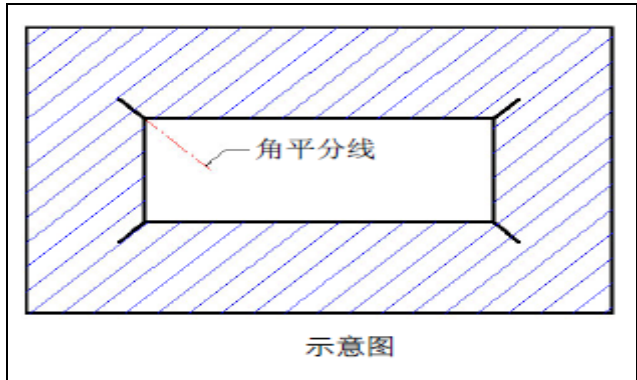
注：如NC 程序中无G34/G35，则缺省为取消减速加工。

5.5.5 G36,G37 (在拐点延长给定距离)

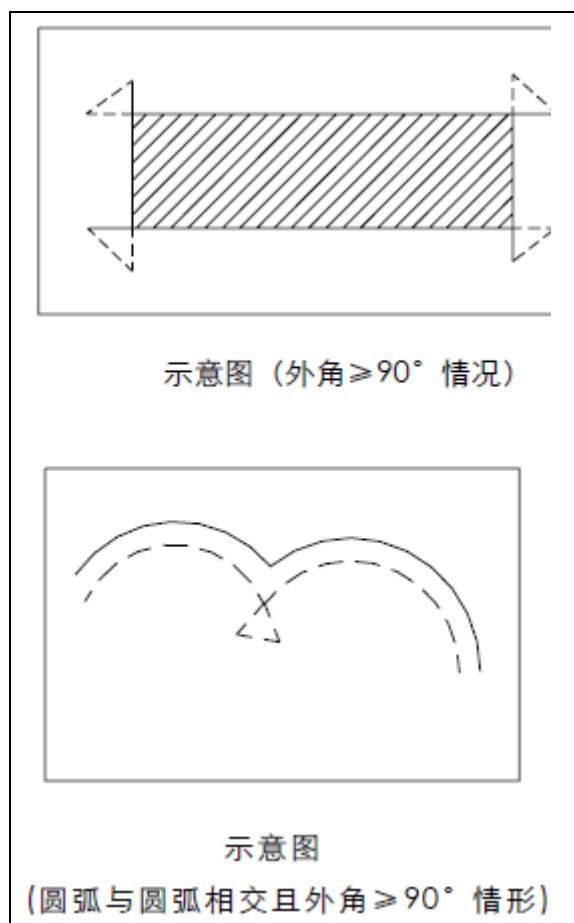
□ G37：为了在G01，G02，G03 的程序段产生过切，但只在两轴并有补偿的程序中起作用。

□ 内角时，沿拐角的角平分线方向向外延

长给定距离（右如图）；



□ 外角时：当外角为钝角时，沿直线端的终点方向或圆弧切线方向延长给定距离；当外角为锐角时，按照尖角直线过渡的方法处理（尖角直线过渡的处理方法见5.10，此时与过切量无关）。



□ G36 取消G37

□ G36, G37 格式：

G36;

G37X__[T__]

注意：1、[]中的内容是可选项，即指令可为：G37X__或G37X__T__；

X 后的数为要延长的距离，该距离为大于零的数，在切内角时要注意延长的距离不能太大，根据精度要求给定延长距离或丝的半径。T 后的数是在过切开始之前及过切过程中每条线段端点处的暂停时间，过切后无暂停。

2、若获得比较好的清角效果，还须其它方面的配合，例如适当增加张力，采用细丝和根据工件高度调整合适的暂停时间[在试验中曾经加工20mm 高度的工件时，采用细丝 $\varnothing 0.15\text{mm}$ 、静张力1250g($\varnothing 0.2\text{mm}$ 丝的配重)，和暂停时间8s（工件越厚，暂停时间越长），获得较好的清角效果]。

建议 $\varnothing 0.12\sim 0.15\text{mm}$ 丝均使用同样的重锤。

G37、G36 应放在程序面上。且若有过切指令G37 时，在退出程序面之前一定要有G36 指令，以取消过切指令。如：

H000=+00000000;

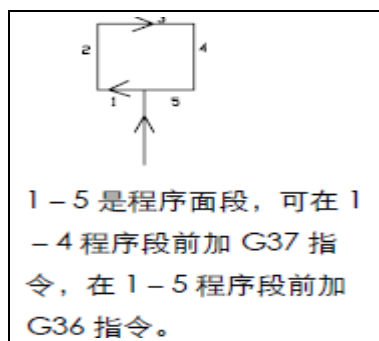
H001=+00000110;

H005=+00000000;

```

T84 T86 G54 G90 G92X+1500Y-1500;
C007;
G01X+1500Y-1000;G04X0.0+H005;
G41H000;
C004;
G41H000;
G01X+1500Y+0;G04X0.0+H005;
G41H001;
G37X0.05T5.; ----->从本段开始，每段线段的终点延长0.05mm 或在终
X+0Y+0;G04X0.0+H005;          点拐角的角平分线向外延长0.05mm(内角按角平分
X+0Y+3000;G04X0.0+H005;        线处理，外角按延长线处理)。
X+3000Y+3000;G04X0.0+H005;
X+3000Y+0;G04X0.0H005;
G36; ----->从本段开始，取消延长
X+1500Y+0;G04X0.0+H005;
G40H000G01X+1500Y-1000;
M00;
C007;
G01X+1500Y-1500;G04X0.0+H005;
T85 T87 M02;

```



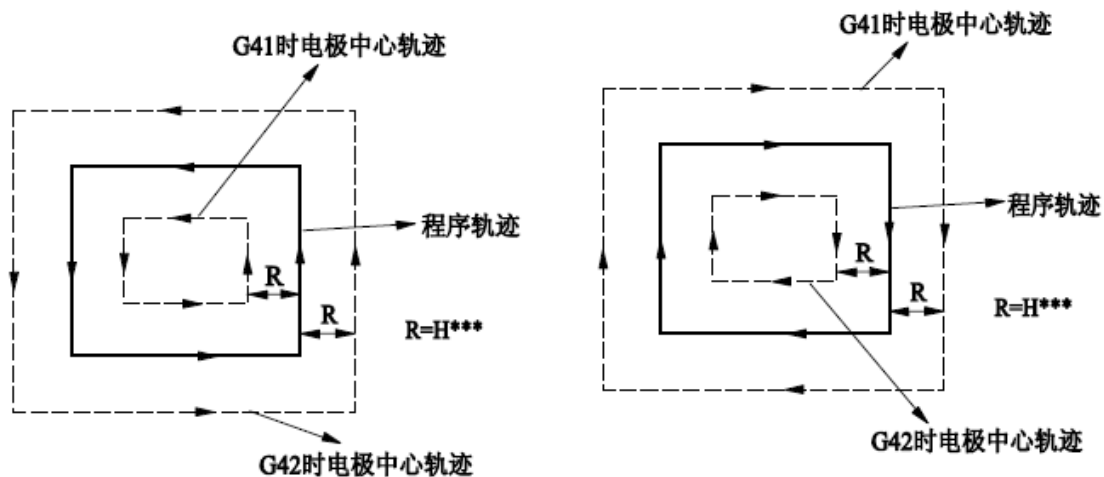
注意：当在NC 程序中增加G37X0 时，则对于直线及圆弧均不进行内角/外角的特殊处理。若NC 程序中有G37X___[T ___]指令时，则自动进行清角处理(X 值必须不为零)，这样所加工工件会有较明显的痕迹，但不影响尺寸精度及粗糙度，其尖角比较好。在NC 程序中如无G37，则用缺省处理方式。

5.6 G40,G41,G42 (补偿和撤消补偿)

电极补偿功能就是电极中心轨迹偏离编程轨迹一定的距离，偏移量的大小等于电极半径加上放电间隙，它可以向电极前进方向的左边(G41)或电极前进方向的右边(G42)进行补偿。

G40	取消电极补偿
G41	电极左补偿
G42	电极右补偿

左补和右补的情形见下图：

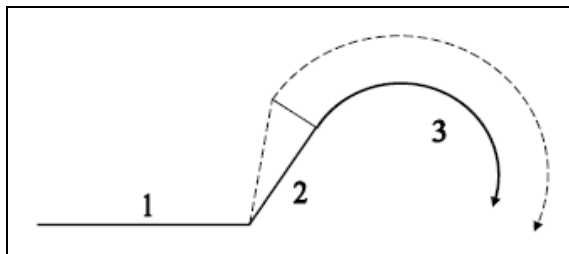


5.6.1 补偿值H

补偿值可以通过三位十进制的补偿值代号来进行指定，即 H^{***} ，每一个补偿号对应一个具体的补偿值，补偿值代号从000~099 共100 种，范围为0.001mm~99999.999mm，用户可以通过： $H^{***} = \underline{\hspace{2cm}}$ 格式为某个补偿号赋值。

5.6.2 补偿开始的情形

从无补偿到有补偿的第一个运动程序段，称为补偿的初始建立段，如左图示：



在第1 段中，无补偿，电极中心轨迹与编程轨迹重合，第2 段中，补偿由无到有，称为补偿的初始建立段，第3 段中，补偿一开始即已存在，故称为补偿进行段。

注意：在补偿初始建立段中，规定运动指令只能是直线插补，不能有圆弧插补指令，否则会出错。

补偿初始建立的各种情形：(以左补偿为例，右补偿同理)。

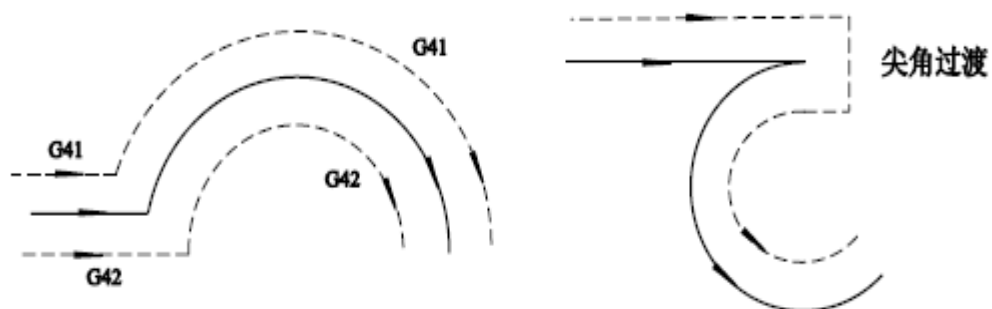


5.6.3 补偿进行中的情形

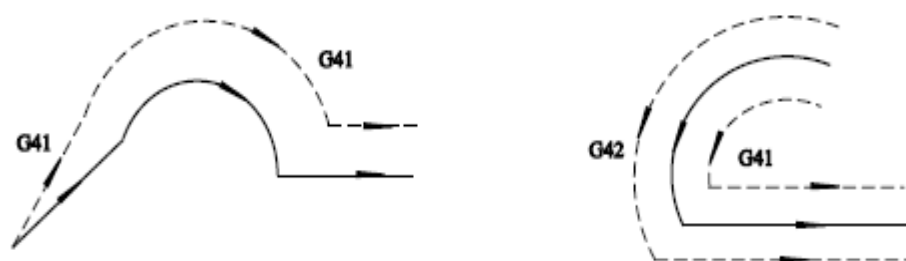
[1] 直线—直线



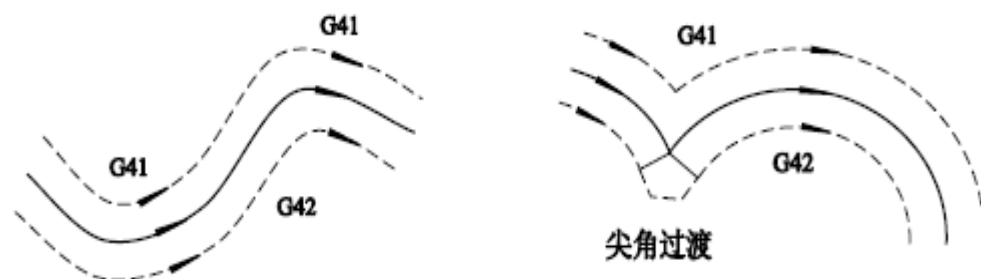
[2] 直线—圆弧



[3] 圆弧—直线



[4] 圆弧—圆弧



5.6.4 补偿撤消时的情形



注意：

[1] 补偿撤消时只能在直线段上进行，在圆弧插补上撤消补偿将会引起错误。

例：G40 G01 X0.; (正确)

G40 G02 X20. Y0 I10. J0; (错误)

[2] 补偿撤消用G40 代码。

[3] 当补偿值为零时，系统会象撤消补偿一样处理，即从电极当前点直接运动到下一个点，但补偿模式并没有被取消。

5.6.5 改变补偿方向

当在补偿方式下改变补偿方向时(由 G41 变为G42, 或由G42 变为G41), 电极由第一段补偿终点插补轨迹直接走到下一段的补偿终点。见下例：

H000=0.;

H001=5.;

G90 G92 X0 Y0;

G41 H001;

G01 X10.Y0.;

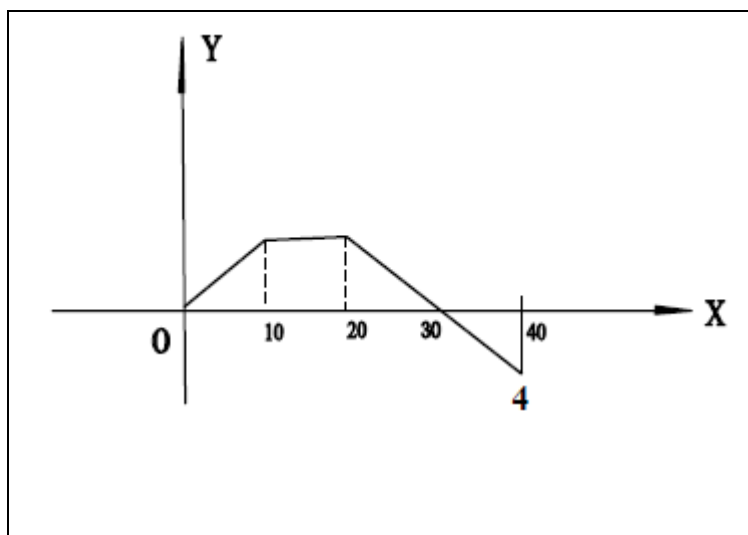
X20.Y0.;

G42 H001;

X40.Y0.;

G40H000G01X40.Y1.;

M02;

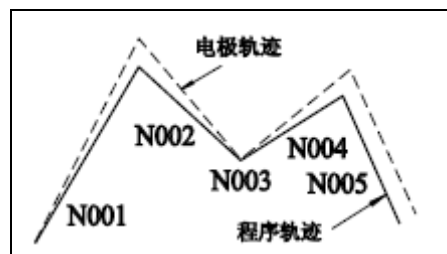


5.6.6 补偿模式下的G92 代码

在补偿模式下，如果程序中遇到了G92 代码，那么补偿会暂时取消，在下段时象补偿起始建立段一样再把补偿值加上，见下例。

例：

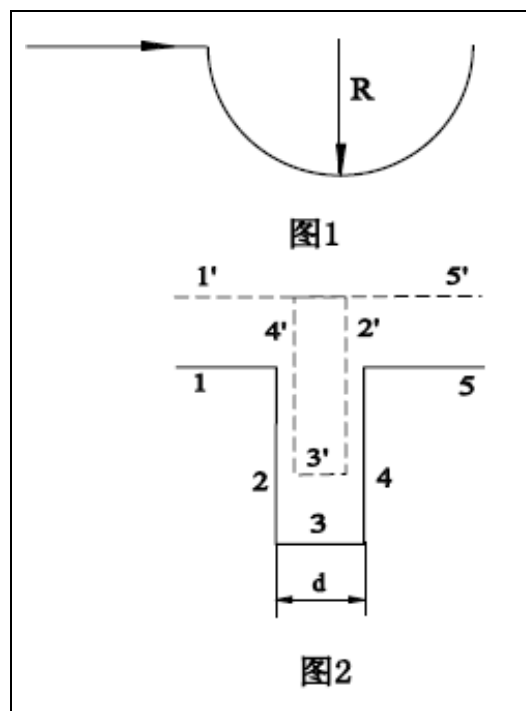
```
N001 G90 G41 H001 G01 X300 Y600;  
N002 X600 Y300;  
N003 G92 X100 Y200;  
N004 G01 X400 Y400;  
N005
```



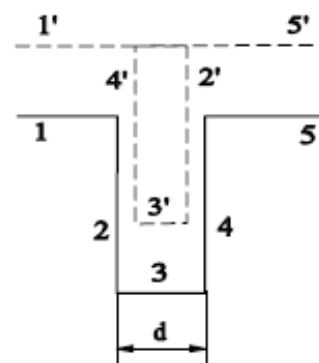
5.6.7 关于过切

当加工轨迹很小，而补偿值过大时就会出现过切，请看下面例子：

[1] 当补偿值大于该圆半径R 时，若选择G41 代码就会发生过切（图1）。



[2] 当补偿值大于槽宽d/2 的情况，也会产生过切。
如图：1-2-3-4-5 为编程轨迹，1' -2' -3' -4' -5' 为补偿
后电极丝中心的运动轨迹，即补偿值大于d/2 就会
发生过切。



[3] 当发生过切时，程序执行将被中断。

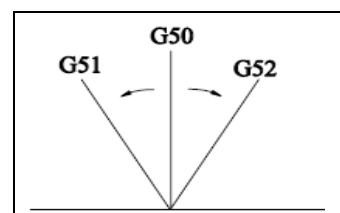
5.7 G50, G51, G52（锥度加工）

所谓锥度加工(Taper 式倾斜加工)，是指电极丝向指定方向倾斜指定角度的加工。

G50 为取消锥度。

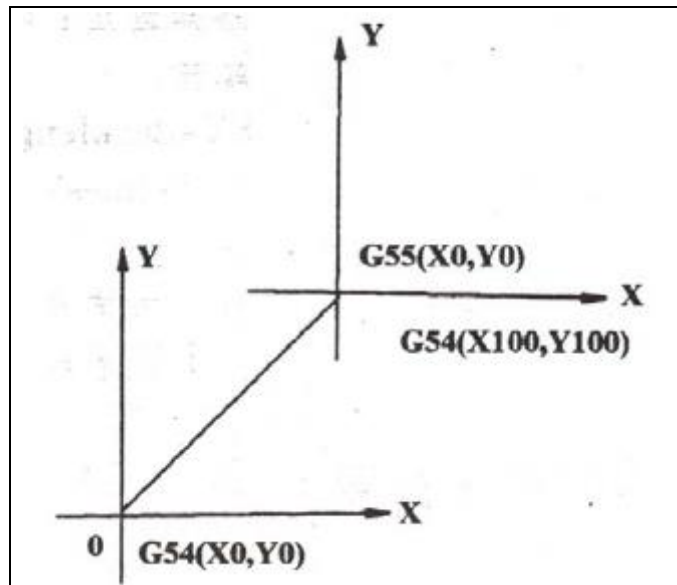
G51 锥度左倾斜(沿电极丝行进方向看，丝向左倾斜)。

G52 锥度右倾斜(沿电极丝行进方向看，丝向右倾斜)。



5.8 G54 G55 G56 G57 G58 G59 (工作坐标系0~5)

这组代码用来选择工作坐标系0~5，共有6个坐标系可被选择，定义坐标系主要是为了编程方便。这组代码可以和G92一起使用，G92代码只能把当前所在坐标系中当前点的坐标定义为某一个值，但不能把这点的坐标在所有坐标系中都定义成该值。



G54	工作坐标系0
G55	工作坐标系1
G56	工作坐标系2
G57	工作坐标系3
G58	工作坐标系4
G59	工作坐标系5

例：G92 G54 X0 Y0 ;
 G00 X100 Y100;
 G92 G55 X0 Y0;

对于上面这个例子，首先把当前点定义为工作坐标系0 的零点，然后X、Y 轴都快速移动100 μ m，并把该点定义为工作坐标系1 的零点。

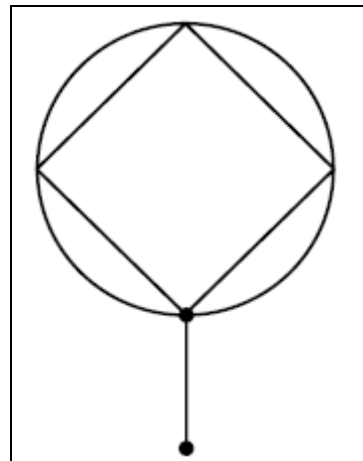
5.9 G60 G61(上下异形)

工件的上面形状和下面形状不同时,根据编程的要求对工件上、下面实施不同形状的加工。
G61 为上、下异形打开, G60 为上、下异形关闭。在上、下异形打开时,不能用G74、G75、G50、G51、G52 等代码。

上面形状代码与下面形状代码的区分符号为 “:”, “:” 左侧为下面形状, “:” 右侧为上面形状。上、

下异形程序举例:

```
H000=0 ;  
H001=0.135;  
G90 G92 X0 Y0 U0 V0;  
C001;  
G61;  
G41 H000;  
G01 X0 Y10. : G01 X0 Y10.;  
G41 H001;  
G02 X-10. Y20. I0 J10. : X-10. Y20.;  
X0 Y30. I10. J0 : X0 Y30.;  
X10. Y20. I0 J-10. : X10. Y20.;  
X0 Y10. I-10. J0 : X0 Y10.;  
G40 H000;  
G01 X0 Y0 : X0 Y0;  
G60;  
M02;
```



5.9.1 G74, G75 (四轴联动)

根据所指定 X、Y、U、V 四个轴的数据，可加工工件上、下面的不同形状。

G74 为四轴联动打开, G75 为四轴联动关闭。

注意: 四轴联动仅支持直线插补代码**G01**。不支持的代码有**G02, G03, G50, G51, G52, G60, G61**。

四轴联动程序举例:

G92 X0 Y-1.;

G74; (四轴联动打开)

G01 Y0;

X10.;

Y10. U-3. V-4.;

X5. U0 V0;

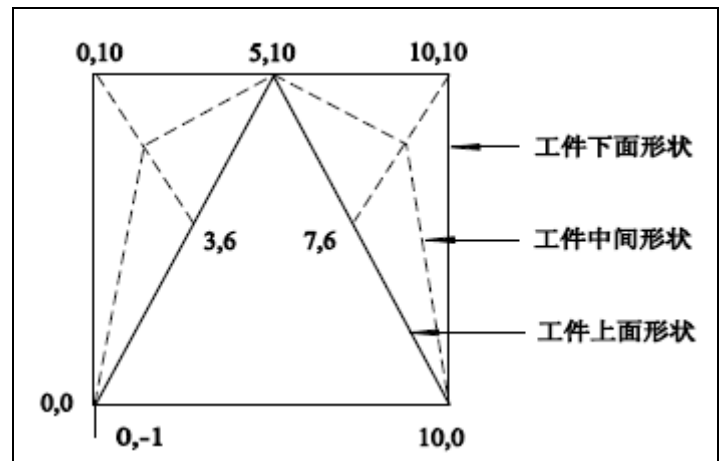
X0 U3. V-4.;

Y0 U0 V0;

G75; (四轴联动关闭)

Y-1.;

M02;



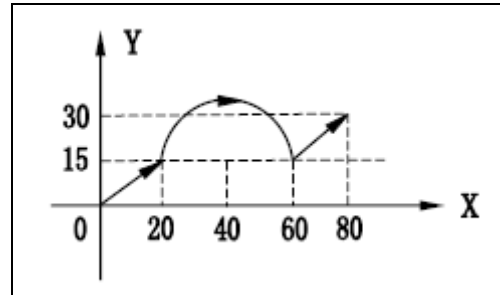
5.10 G90(绝对坐标指令)、G91 (增量坐标指令)

G90: 绝对坐标指令，使用此代码后，所有坐标值都应以绝对方式进行输入，即坐标值是以工作坐标系的零点为参考点计算的。

G91: 增量坐标指令，使用此代码后，所有坐标值的输入都应以增量方式进行，即当前点的坐标是相对于上一个点的，即以上一个点为参考点

例1：绝对坐标的示例：

```
N0001 G90 G92 X0 Y0 ;  
N0002 G01 X20. Y15.;  
N0003 G02 X60. Y15. I20. J0;  
N0004 G01 X80. Y30.;
```



例2：改写例1 程序为增量坐标：(图形同例1)

```
N0001 G91 G92 X0 Y0 ;  
N0002 G01 X20. Y15.;  
N0003 G02 X40.0 I20.;  
N0004 G01 X20. Y15.;
```


5.11 G92(设置当前点的坐标值)

G92 代码把当前点的坐标设置成你需要的值。

例1: G92 X0 Y0 U0 V0;.....把当前点的坐标设置为(0,0,0,0)。

例2: G92 X10 Y0;.....把当前点的坐标设置为(10, 0)。

注意:

[1] 在补偿方式下, 如果遇到**G92** 代码, 会暂时中断电极半径补偿功能, 即每执行一次**G92**, 相当于撤消一次补偿, 执行下一段程序时, 再建立一次补偿。

[2] 每个程序中一定要有**G92** 代码, 否则可能会发生不可预测的错误。

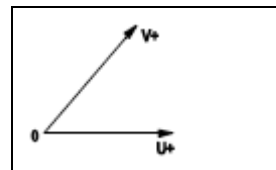
6 X、Y、U、V (I、J) 坐标轴

所谓坐标轴就是能使工作台和电极丝产生相对移动的部件。在编程中它是一个字, 由一个移动坐标轴的地址(字母)和它后面的数字组成, 数字表示一个坐标轴的运动量。它可以用绝对和增量的方式进行指定。

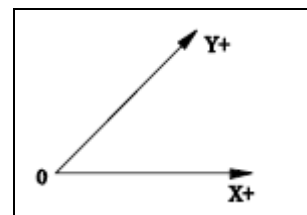
坐标轴和它的方向由下列方法进行确定:

面对机床

- ☐ 电极丝向工作台右边作相对运动(实际为工作台向左运动)为X轴正向, 向工作台左边作相对运动为X轴负向;
- ☐ 前后方向为Y轴, 电极丝移向操作者为Y轴负向, 远离操作者为Y轴正向;



- ☐ 与X轴平行的为U轴, 上导丝嘴向右移动为U轴正向。
- ☐ 与Y轴平行的轴为V轴, 上导丝嘴远离操作者为V轴正向。



I、J 是圆弧插补时的指令参数, 圆心相对于圆弧起始点坐标对应于X、Y轴分别用I、J来表示, 它后面的值用增量方式来表示。

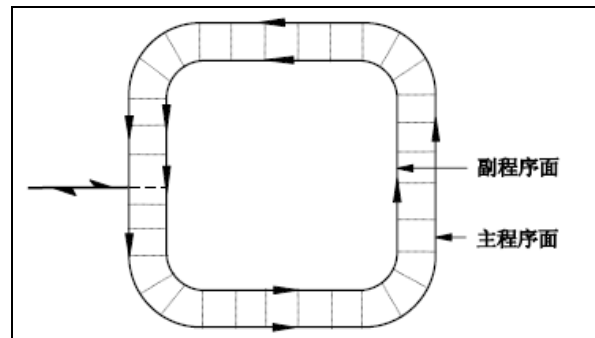
所有数字的输入(坐标值)由于所用的计量单位不同而有所不同, 请看下表:

计量单位	单位	最大命令值	最小命令值
公制	0.001mm	999.999mm	0.001mm
英制	0.0001 英寸	99.999 英寸	0.0001 英寸

7 锥度加工

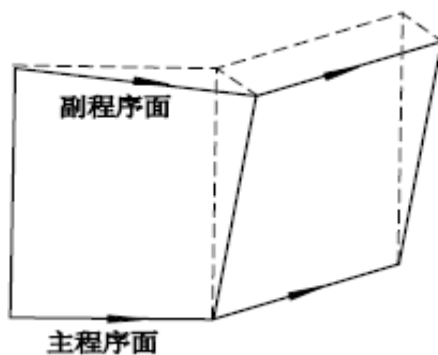
7.1 锥度加工实例

```
H000=0;  
G90 G92 X-18.Y0.U0.V0.;  
C001;  
G42H000;  
G52A0.;  
G01 X-10.Y0.;  
G42H000;  
G52A2.5;  
X-10.Y-6.;  
G03X-6.Y-10.I4.J0.;  
G01X6.Y-10.;  
G03X10.Y-6.J4.;  
G01X10.Y6.;  
G03X6.Y10.I-4.;  
G01X-6.Y10.;  
G03X-10.Y6.J-4.;  
G01X-10.Y0.;  
G40H000G50A0.X-18.Y0.;  
M02;
```

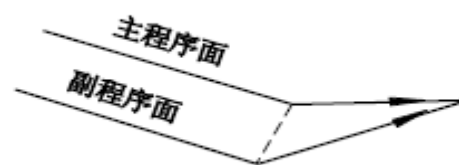
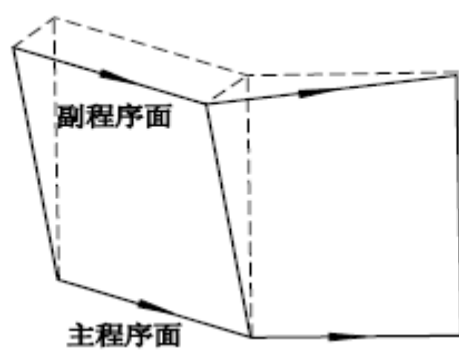


7.2 锥度加工的开始

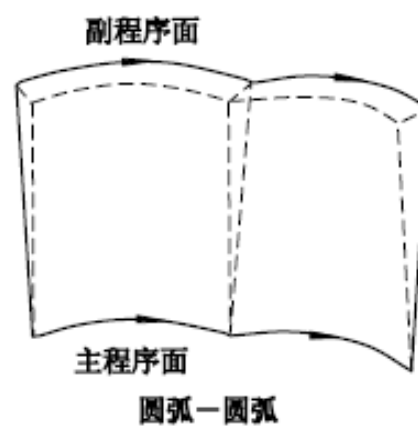
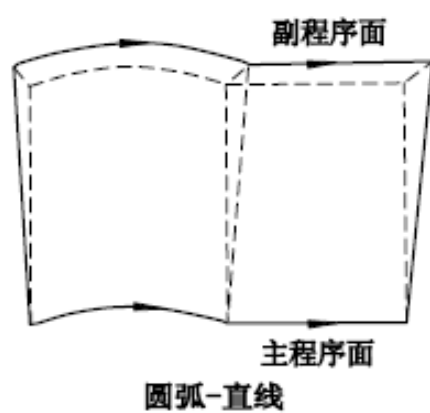
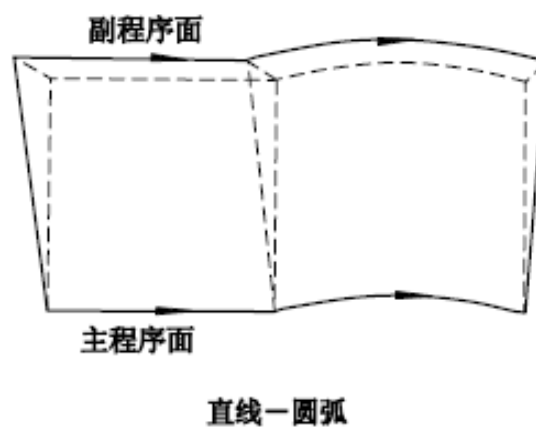
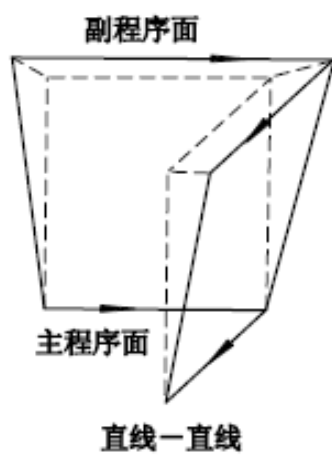
锥度加工开始时的动作如下图所示，与偏置量补偿一样，不能用圆弧指令（G02, G03）来开始。



锥度加工结束时的动作如下图所示：



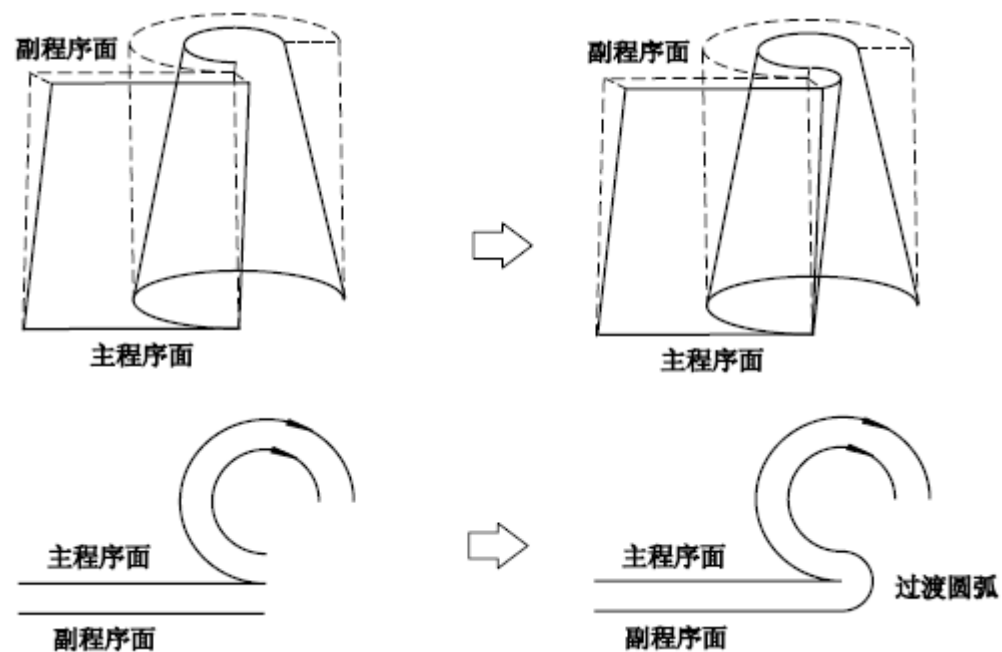
一般的锥度加工如图所示：



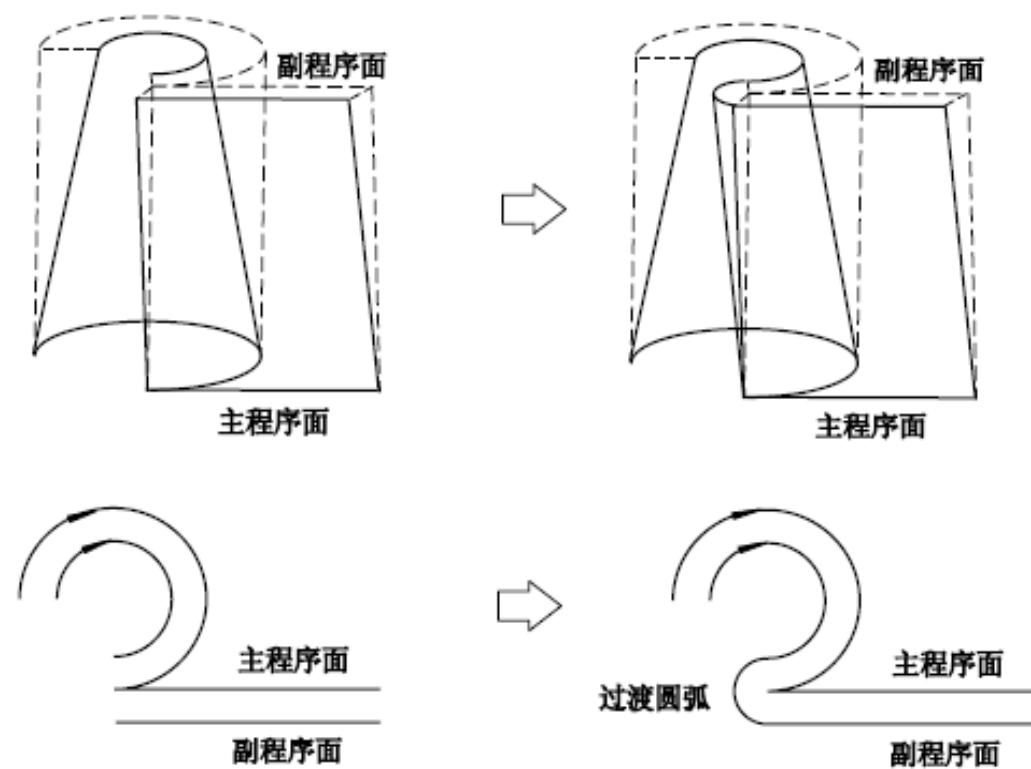
7.3 锥度加工的连接

在锥度加工过程中，工件副程序面两曲线将自动插入过渡圆弧。

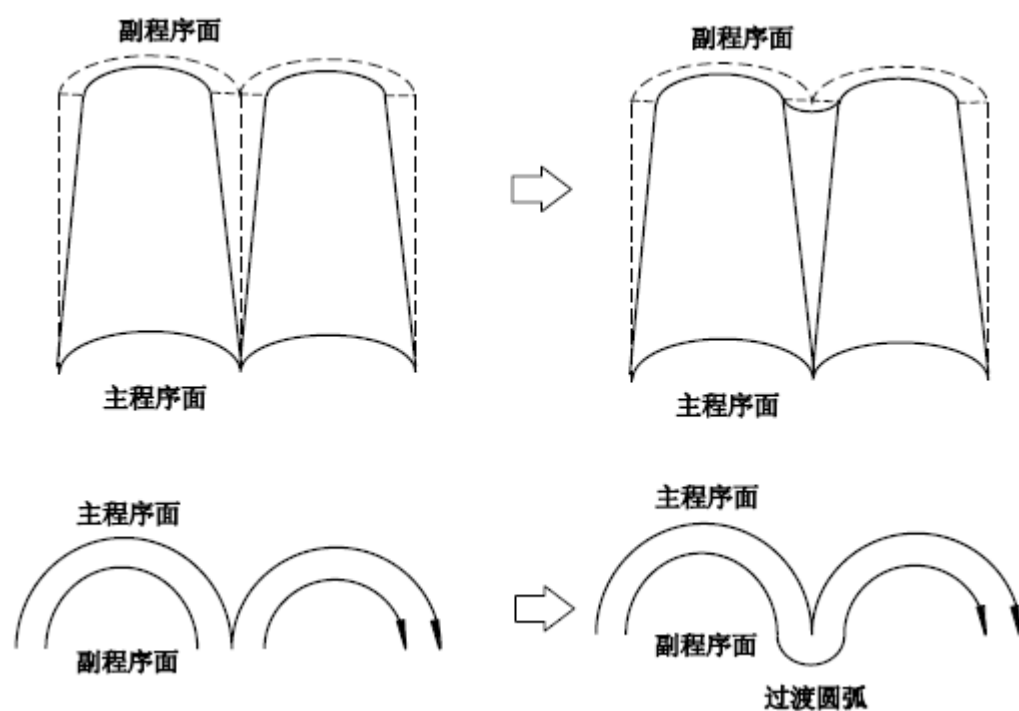
[1] 直线—圆弧



[2] 圆弧—直线



[3] 圆弧—圆弧



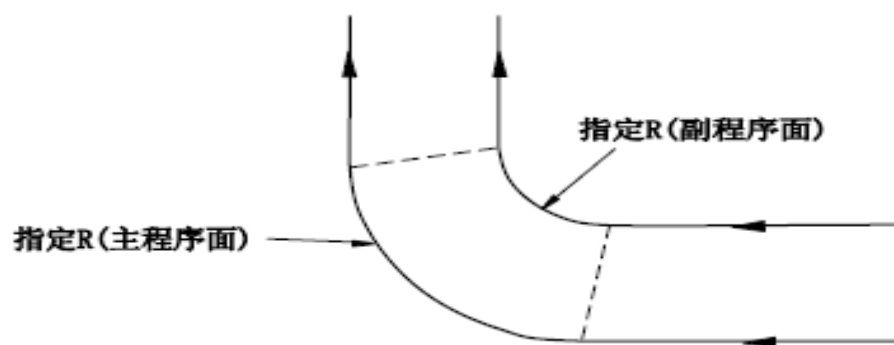
7.4 锥度和转角R

在锥度加工中，可以在主程序面和副程序面分别加入圆弧过渡，方法是：在该程序段加入转角R 指令，用第一个R 设定主程序面的过渡圆弧半径，用第二个R 设定副程序面的过渡圆弧半径，格式如下：

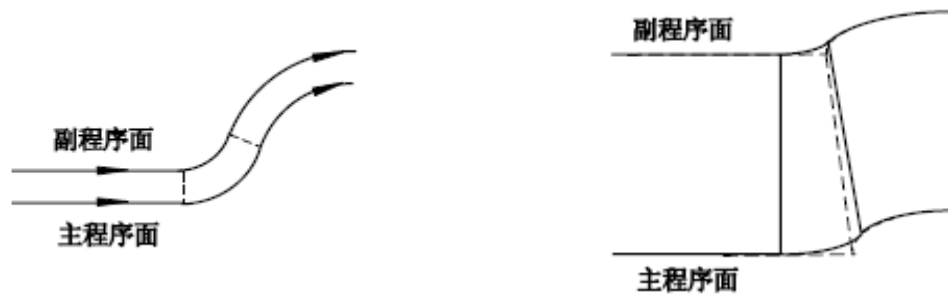
G01 X__ Y__ R__ R__;

G02 X__ Y__ I__ J__ R__ R__;

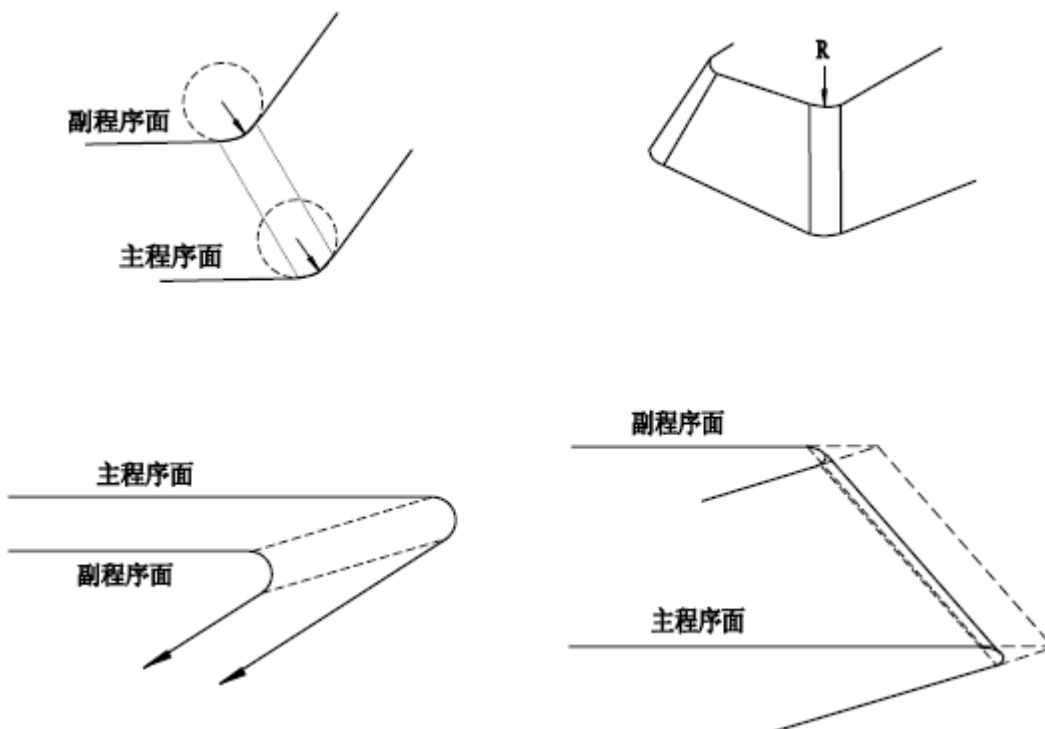
G03 X__ Y__ I__ J__ R__ R__;



锥度加工加入圆弧过渡如下图所示：



如果两个R 相等，则工件的上、下面插入同一圆弧，因而成斜圆柱状，如下图所示：



根据此指令与下一分段的移动指令，指定了半径的转角R 被插入，此时必须是在补偿状态（G41, G42）且锥度状态（G51, G52）。若补偿、锥度都处于取消状态（G40, G50）时，转角R 不被插入。

7.5 恒锥度和变锥度

恒锥度或变锥度的开始及取消必须用直线， 现举例说明如下：

```
H000=0;
G90G92X0.Y0.U0.V0.;
C096;
G01X0.Y4.;
C001;
G42H000;
G51A0G01X0.Y5.;           从引入线预加锥度。
G42H001;
G51A2.X5.Y5.;             从本段开始立即加上2.0 度锥度。
G51A2.X5.Y-5.;
G51A2.X-5.Y-5.;
G51A0.X-5.Y5.;           改变目前的锥度角为0 度。
G51A2.X0.Y5.;            改变目前的锥度角为2.0 度。
G40H000G50A0X0.Y4.5;     从本段开始立即取消2.0 度锥度。
X0.Y0.;
M02;
```

注意若这时在绘图或执行时如出现“圆弧-圆弧或直线-圆弧处理错误”提示，则说明在该处无法实现锥度的改变。

8 M 代码

8.1 M00 (暂停指令)

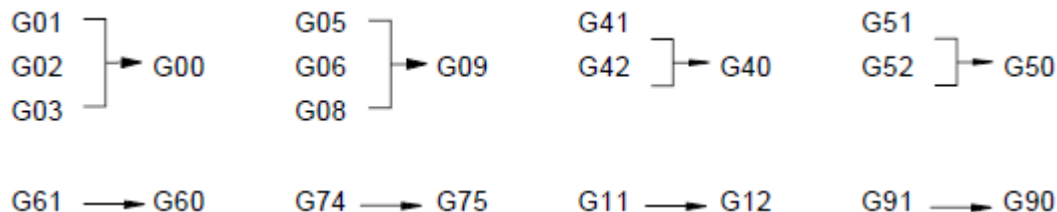
执行M00 代码后，程序执行暂停。按启动键后，程序接着执行。

```
例如： G000X50.;
      M00;
      C001
      G01 X10.;
```

8.2 M02(程序结束)

M02 代码是整个程序结束命令，M02 之后的代码将不被执行。执行M02 代码后，系统将复位所有的延续至程序结束的模态代码的状态，然后再接受用户的命令以执行相应的动作。

要复位的代码和复位后状态如下：



也就是说上一个程序中出现过的以上代码不会对下一个要执行的程序构成影响，除非你在下一个程序中使用了以上代码。例如你当前执行的程序中有G05 代码，而在M02 之前又无G09 代码，则执行下一个程序时X 轴镜像不起作用。M02 能把G05 自动恢复成G09 状态，除非你在下一个程序中又用了G05 代码。

8.3 M98(子程序调用)

M98 代码用来指定要调用的子程序号。

格式：M98 P**** L***；

关于此代码更详细的描述请看“子程序”一节。

8.4 M99(子程序结束)

M99 代码表示一个子程序结束，它是子程序的最后一个程序段，当执行此代码后，程序重新返回到主程序中，并执行下一个程序段。

8.5 M50(剪丝代码)

M50 表示某一程序段加工结束后，进行机动或手动剪丝的代码。

8.6 M60(穿丝代码)

M60 表示多孔加工时某一孔位加工结束后，在移动至另一孔位时进行机动或手动穿丝的代码。

9 C 代码

C 代码是用来在程序中选择加工条件代号的代码。格式为“C”后跟三位十进制数，“C”和数字

之间不能有别的字符。例如：C001。

当程序执行这条指令时，加工条件C001 的各个参数将显示在CRT 的加工条件显示区域中。如果程序中所选的加工条件号在加工条件表中不存在，将会产生一个错误。

注意：地址C 后所跟的数字必须是三位十进制数，不够三位请用“0”补齐，否则将会出错。

10 子程序

有时，在同一个程序中，相同程序会多次出现，如果把这些相同的程序放在一个固定的程序中，在需要的地方用这个固定程序代替，那么在减少程序的复杂性和长度方面将会收到很好的效果。

我们把这个固定的程序叫子程序，引用此固定程序的程序叫主程序。由于程序中可以使用顺序号(N****)，所以为子程序的调用提供了可能，我们通过在一个程序的开始置一个顺序号来定义此子程序。当在主程序中要调用时，只需指定调用子程序的顺序号即可。调用一个子程序时，此子程序将被当作一个单段程序对待。一个调用子程序代码只能调用一个子程序，子程序中还可以调用别的子程序，子程序号最大为9999。一个子程序以M99 代码作为结束标志。当执行到M99 代码时，程序返回到主程序，并接着执行下一段程序，子程序的格式如下：

N****;

程 序

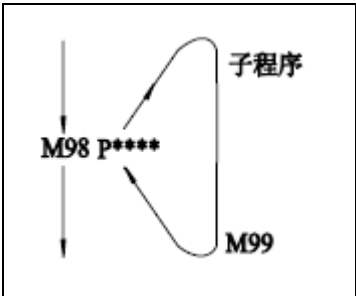
M99 ;

主程序调用子程序的格式为：

M98 P**** L*** 其中：

P**** 为要调用的子程序的顺序号

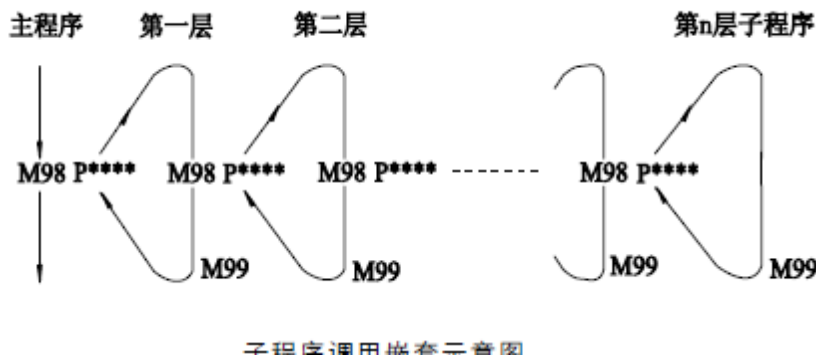
L*** 为调用子程序的次数。



如果L*** 省略，此子程序只被调用一次，当为“L0”时，将不调用此子程序。地址“L”后可跟

一至三位十进制数，但是一个子程序一次最多可调用999 次。例如：L5 等。

在一个子程序中也可以调用别的子程序，它的处理和主程序中调用子程序相同，主程序调用子程序，子程序中再调用子程序，这称作嵌套(nesting)，如下图所示：



在本系统中规定n 的最大值为9，即子程序嵌套最多为九层。

11 关于运算

本系统支持的运算符有：
+, -, dH***..... (dxH***), d 为一位十进制数

11.1 运算符地址

在式子中(地址后所接代码、数据)能够用运算符的地址如下：

种类	地址
坐标值	X, Y, U, V, I, J
赋值类	H

11.2 优先级

所谓优先级即执行的先后顺序。
本系统中运算符的优先级如下：

dH***, ↑ 高
+, - ↓ 低

11.3 运算式的书写

运算符的式长只能在一个段内。
例1：H000=1000 ；
G90 G01 X1000+2H000;

则X 轴直线插补到3000 μm 处。

例2：H000=320 ；
H001=180+2H000 ；.....H001 等于820

12 H 代码(补偿)

H 代码实际上是一种变量，每个H 代码代表一个具体的数值，在程序中用赋值语句对其进行赋值，如下所示：

H000=500.001 ；
一个H 代码的格式如下：
H***

即地址“H”后接三位十进制数，不够三位的请用“0”补齐。例：H010
H 代码有H000~H099 共一百种，每个H 变量的赋值范围为±99999.999 mm。

在程序中 H 代码可以当作变量被引用。

例：H005=90.07；
G01 X50.1+5H005；
则X 轴以直线插补方式运动到
50.1+5×90.07=500.45mm 处。
(对H 代码可以作：+、-和倍数运算)

例：H100=H010+10-2H000；
若H010=100，H000=20
则H100=100+10-2×20=70

13 代码的初始设置

有些功能的代码在如下一些情况下要回到初始设置状态：

- [1] 刚打开电源开关时；
- [2] 执行中遇到程序结束指令M02 时；
- [3] 在程序执行期间按下了结束钮时；

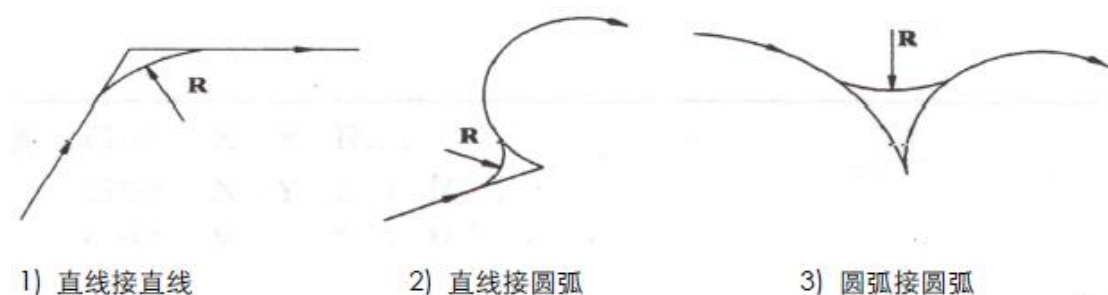
要回到初始设置状态的代码和它们的初始值如下：

G-代码类：

初始状态	可设置的状态
G00	G01
	G02
	G03
G09	G05
	G06
	G08
G12	G91
G40	G41
	G42
G50	G51
	G52
G60	G61
G75	G74
G90	G91

14 转角功能

转角R 功能即在两条曲线的连接处中加一段圆弧，如下图所示：



圆弧的半径在程序中通过R 来指定，这段圆弧和两条曲线都相切，在程序中用下面的格式来指定转角R 功能：

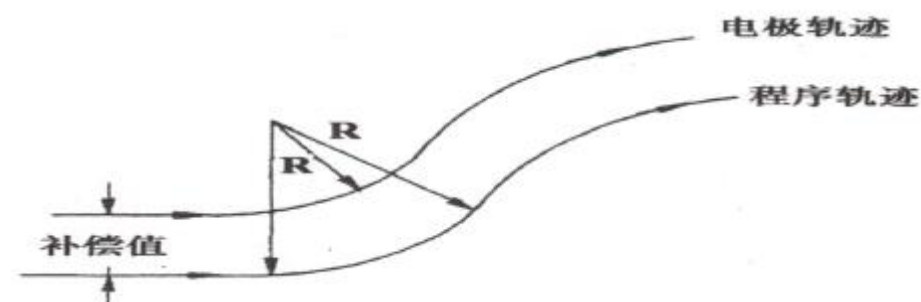
[1] 为了指定转角R 功能，请在要插入转角的第一段曲线的运动代码后，加一个地址“R”，后接圆弧半径值，地址R 和它后面的半径值必须和第一条曲线的运动代码在同一个段内。

例：G01 X__Y__R__ ；
G02 X__Y__I__J__R__ ；
G03 X__Y__I__J__R__ ；

上面这些指令的含义是在本段程序的运动曲线和下一段程序的运动曲线之间加一段半径为R__的圆弧。转角R 功能只有在有补偿的状态下(G41, G42)才会执行，当补偿撤消时，转角R 功能将被忽略，因而当G40 撤消补偿后，程序中虽然有转角R 指定，但并不在两条曲线间加一圆弧，而是按编程轨迹运动，在H***=0 的情形下，转角R 功能有效。

[2] 在G00 代码后加转角R 功能，将不执行转角功能的动作。

[3] 对有补偿量的情形，如下图所示。



15 T代码

T84 代码为打开喷液指令，使加工液由上下导丝嘴喷出，该代码在程序中应放在加工代码之前，以免在加工中由于未冲液而烧断丝，如下例。

例：

```
G92 X0 Y0;
T84;..... 打开喷液
T86;..... 启动走丝
C007
H000=110;
G41 H000;
G01 X5.0;..... 加工指令
C021;
G01 X15.;
Y10.;
X5.;
Y0;
G40 G01 X0;
T85 T87;..... 关闭喷液, 停止走丝
M02;
```

T85 代码为关闭喷液指令，使加工液停止喷出。

T86 代码为启动走丝电机指令，使丝在走丝（启动/停止走丝）机构上高速运转，该代码在程序中应放在加工代码之前，以免在加工中由于丝在同一个地方持续放电而烧断丝，如上例。

T87 代码为停止走丝指令，使走丝电机停止运转。