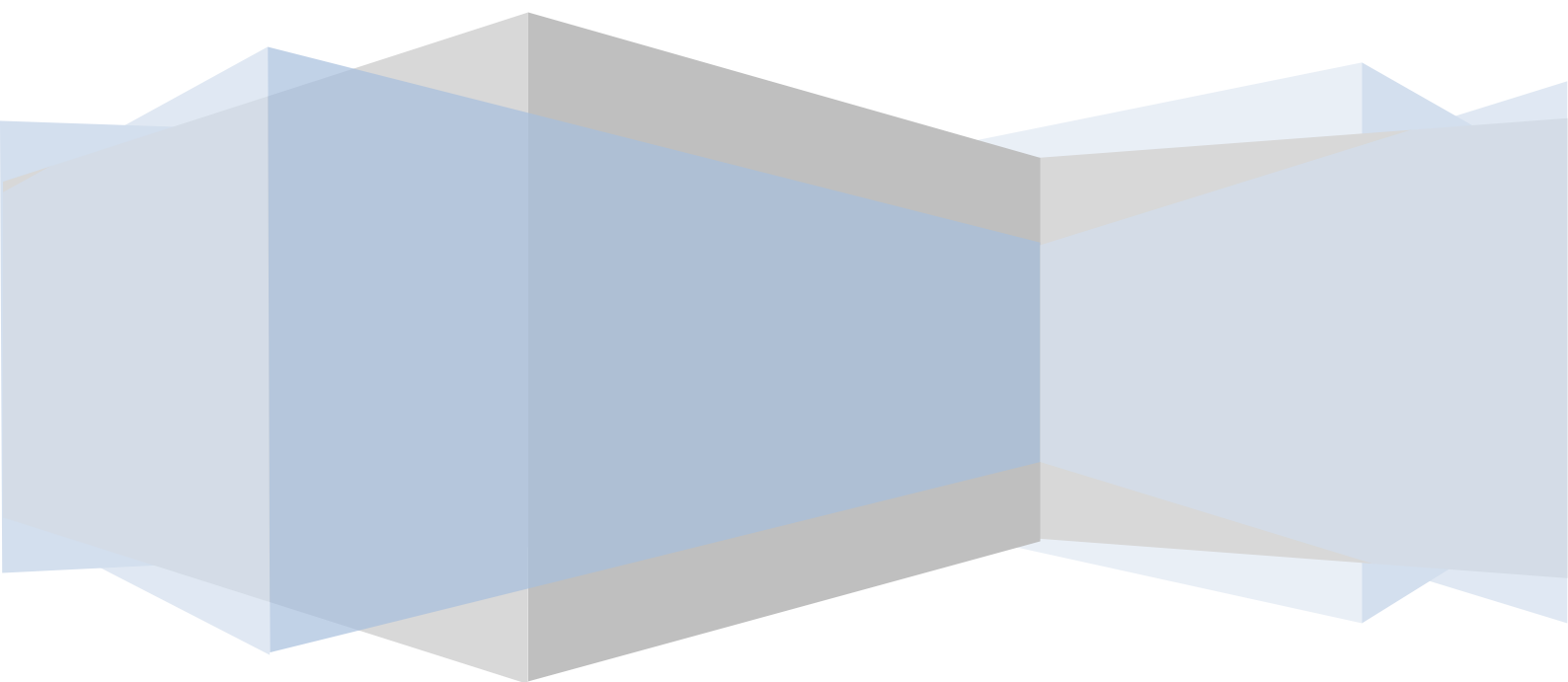


车床程式制作说明书

A-Type



1、G代码指令说明

功 能 名 称	G 代 码	页数索引
快速定位	G00	3
直线、斜线插补	G01	4
圆弧插补（顺时针）	G02	5
圆弧插补（逆时针）	G03	5
暂停	G04	9
圆柱插补	G07.1	10
启动极坐标插补	G12.1	12
取消极坐标插补	G13.1	12
设定 X-Y 工作平面	G17	13
设定 Z-X 工作平面	G18	13
设定 Y-Z 工作平面	G19	13
外径/内径车削循环	G90	14
螺纹车削循环	G92	17
端面车削循环	G94	20
参考点复归	G28	23
跳越机能	G31	24
螺牙切削	G32	25
Z 向攻牙循环	G33	28
变螺距螺纹切削	G34	29
X 轴自动刀具补偿测量	G36	30
Z 轴自动刀具补偿测量	G37	30
刀具半径补偿取消	G40	32
刀具半径左补偿	G41	32
刀具半径右补偿	G42	32
多边形切削取消	G50.2	41
多边形切削	G51.2	41
工件坐标系设定	G54~G59	42
单一宏程序呼叫	G65	43
模式宏程序呼叫	G66	44
模式宏程序呼叫取消	G67	44
英制单位加工	G20	45
公制单位加工	G21	45
精车削循环	G70	46
横向（外径）粗车削循环	G71	47
径向（端面）粗车削循环	G72	51
成形轮廓粗车削循环	G73	55
端面（Z 轴）啄式加工循环	G74	59
横向（X 轴）啄式加工循环	G75	62
复合型螺纹切削固定循环	G76	65
钻孔循环取消	G80	68

端面钻孔、点钻循环	G82	68
端面排屑钻孔循环	G83	71
端面刚性攻牙循环	G84	73
端面搪孔循环	G85	75
侧面钻孔、点钻循环	G86	68
侧面排屑钻孔循环	G87	71
侧面刚性攻牙循环	G88	73
侧面搪孔循环	G89	75
坐标系设定/主轴最高转速限制	G50	76
等表面切削速度	G96	77
等表面切削速度取消	G97	77
每分钟进给量 (mm/min)	G98	79
每转进给量 (mm/rev)	G99	79
附加轴 A, B, C 轴指定增量方式	G91	80

2、M 码说明-----81 页

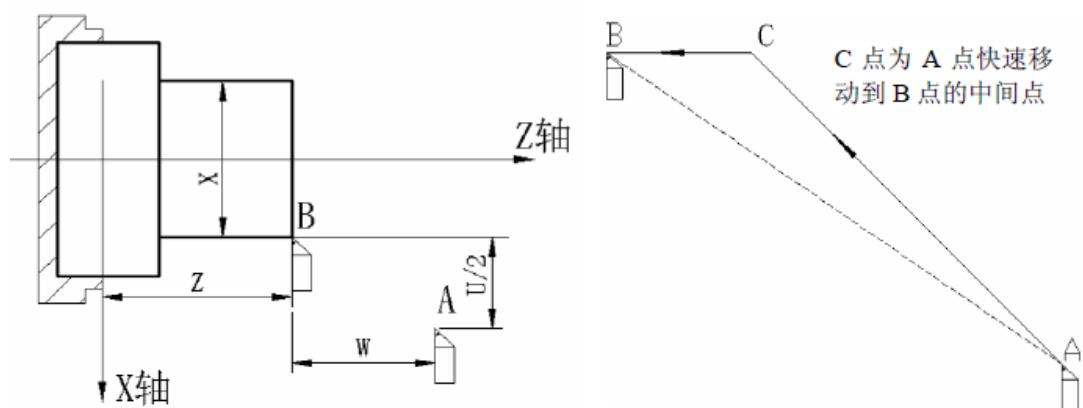
1.2.1 G00（快速定位）

指令格式: G00 X(U)_ Y(V)_ Z(W)_ ;

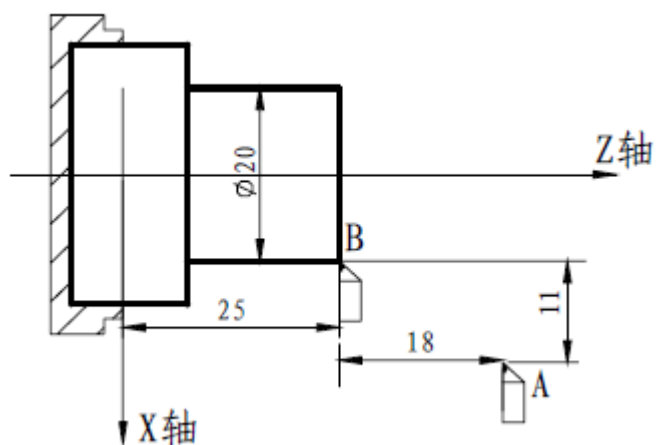
代码功能: X、Y、Z轴同时从起点以各自的快速移动速度移动到终点，三轴是以各自独立的速度移动，短轴先到达终点，长轴独立移动剩下的距离，其合成轨迹不一定是直线。X、Y、Z轴各自快速移动速度分别由系统数据参数007、008、009号设定，实际的移动速度可通过机床面板的快速倍率键进行修调。

代码说明: G00为01组G代码的初值; X(U)、Z(W)可省略一个或全部，当省略一个时，表示该轴的起点和终点坐标值一致; 同时省略表示终点和始点是同一位置，X与U、Z与W在同一程序段时X、Z有效，U、W无效。

运动轨迹图:



示例: 刀具从A点快速移动到B点



G0 X20 Z25; (绝对坐标编程)
 G0 U-22 W-18; (相对坐标编程)
 G0 X20 W-18; (混合坐标编程)
 G0 U-22 Z25; (混合坐标编程)

1.2.2 G01（直线插补）

指令格式：G01 X(U)_ Y(V)_ Z(W)_ F_;

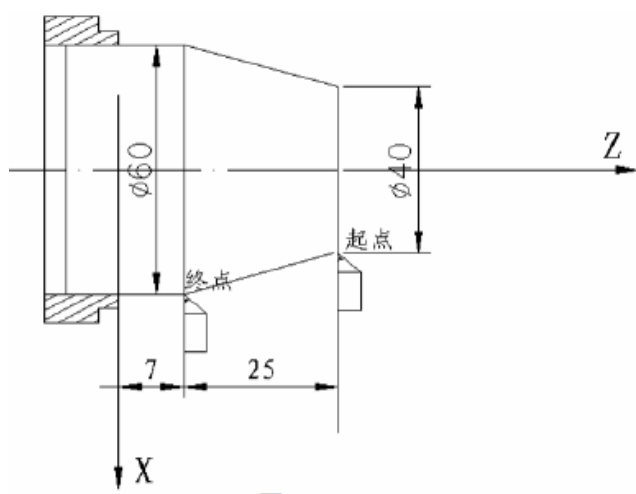
代码功能：运动轨迹为从起点到终点的一条直线。

代码说明：G01为模态G代码；

X(U)、Z(W)可省略一个或全部，当省略一个时，表示该轴的起点和终点坐标值一致；同时省略表示终点和始点是同一位置。F 代码值为X 轴方向和Z 轴方向的瞬时速度的向量合成速度，实际的切削进给速度为进给倍率与F 代码值的乘积；

F 代码值执行后，此代码值一直保持，直至新的F 代码值被执行。后述其它G 代码使用的F 代码字功能相同时，不再详述。

示例：从直径 $\Phi 40$ 切削到 $\Phi 60$ 的程序代码。



程序：

```
G01 X60 Z7 F500; （绝对值编程）  
G01 U20 W-25; （相对值编程）  
G01 X60 W-25; （混合编程）  
G01 U20 Z7; （混合编程）
```

1.2.3 G02/03（圆弧插补）

指令格式： G02/03 X(U)_ Z(W)_ R_
G02/03 X(U)_ Z(W)_ I_ K_

代码功能: G02代码运动轨迹为从起点到终点的顺时针(后刀座坐标系)/逆时针(前刀座坐标系)圆弧, 轨迹如图1所示。G03代码运动轨迹为从起点到终点的逆时针(后刀座坐标系)/顺时针(前刀座坐标系)圆弧, 轨迹如图2所示。

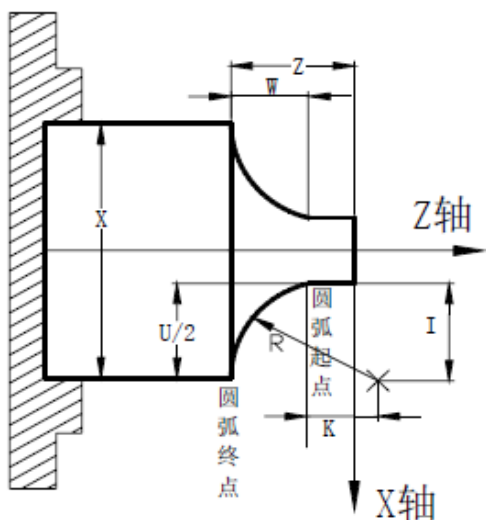


图1

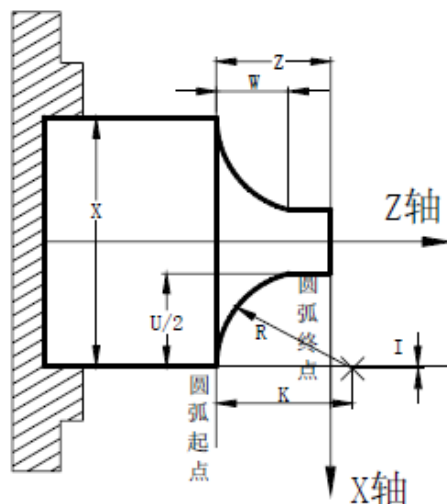


图2

代码说明: G02、G03为模态G代码。

R为圆弧半径;

I为圆心与圆弧起点在X方向的差值, 用半径表示;

K为圆心与圆弧起点在Z方向的差值;

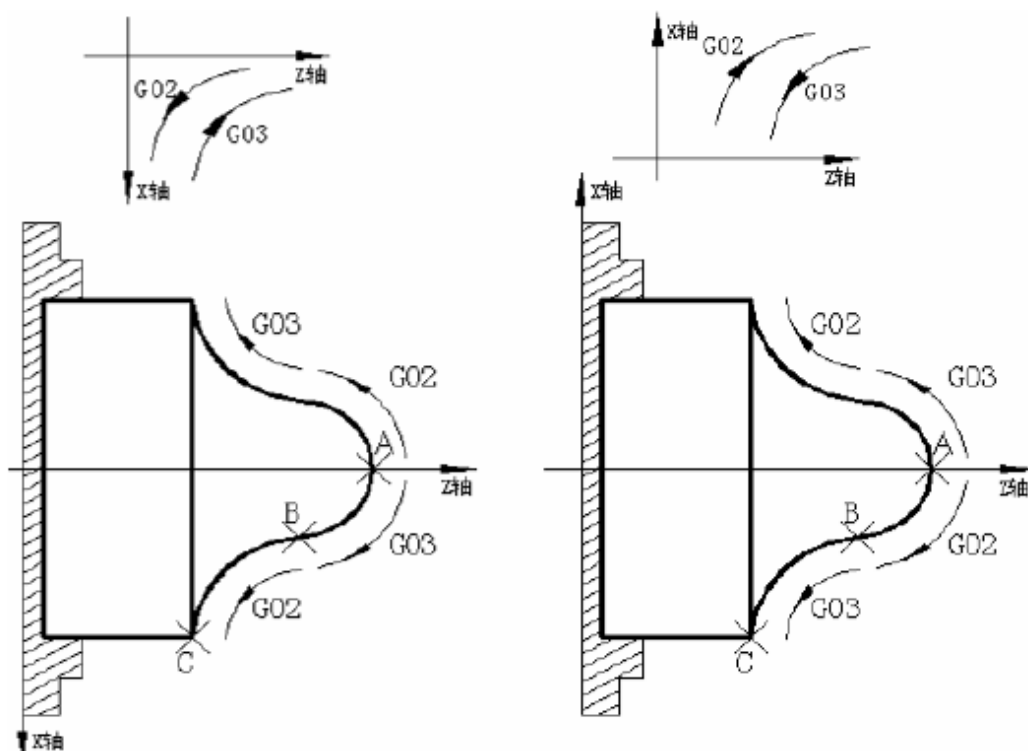
圆弧中心用地址I、K指定时, 其分别对应于X、Z轴I、K表示从圆弧起点到圆心的向量分量, 是增量值;

$I = \text{圆心坐标} X - \text{圆弧起始点的} X \text{坐标};$

$K = \text{圆心坐标} Z - \text{圆弧起始点的} Z \text{坐标};$

I、K根据方向带有符号, I、K方向与X、Z轴方向相同, 则取正值; 否则, 取负值。

圆弧方向: G02/ G03圆弧的方向定义, 在前刀座坐标系和后刀座坐标系是相反的, 见图



注意事项:

当 $I = 0$ 或 $K = 0$ 时, 可以省略; 但地址 I 、 K 或 R 必须至少输入一个, 否则系统产生报警;
 I 、 K 和 R 同时输入时, R 有效, I 、 K 无效;

R 值必须等于或大于起点到终点的一半, 如果终点不在用 R 定义的圆弧上, 系统会产生报警;

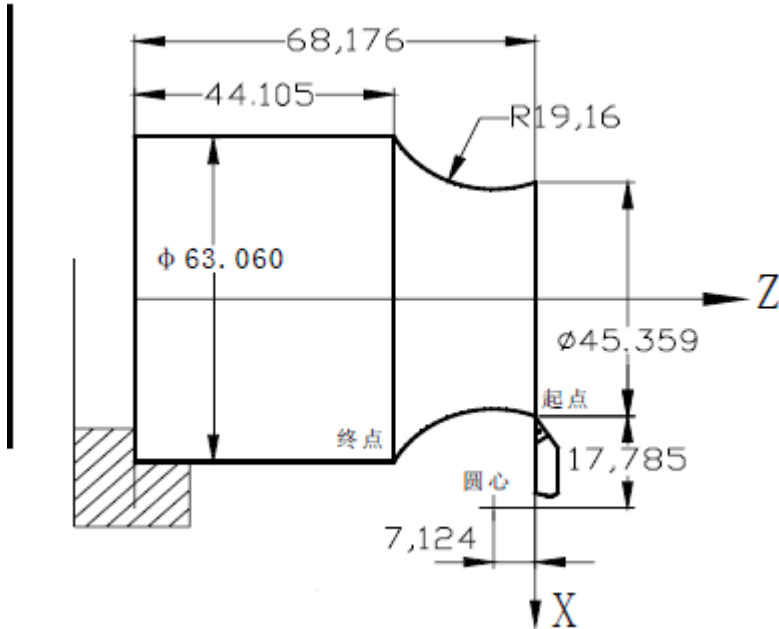
地址 $X(U)$ 、 $Z(W)$ 可省略一个或全部; 当省略一个时, 表示省略的该轴的起点和终点一致; 同时省略表示终点和始点是同一位置, 若用 I 、 K 指定圆心时, 执行 $G02/G03$ 代码的轨迹为全圆(360°); 用 R 指定时, 表示 0° 的圆;

建议使用 R 编程。当使用 I 、 K 编程时, 为了保证圆弧运动的始点和终点与指定值一致, 系统按半径 $R = \sqrt{I^2 + K^2}$ 运动;

若使用 I 、 K 值进行编程, 若圆心到的圆弧终点距离不等于 $R (R = \sqrt{I^2 + K^2})$, 系统会自动调整圆心位置保证圆弧运动的始点和终点与指定值一致, 如果圆弧的始点与终点间距离大于 $2R$, 系统报警。

R 指定时, 是小于 360° 的圆弧, R 负值时为大于 180° 的圆弧, R 正值时为小于或等于 180° 的圆弧;

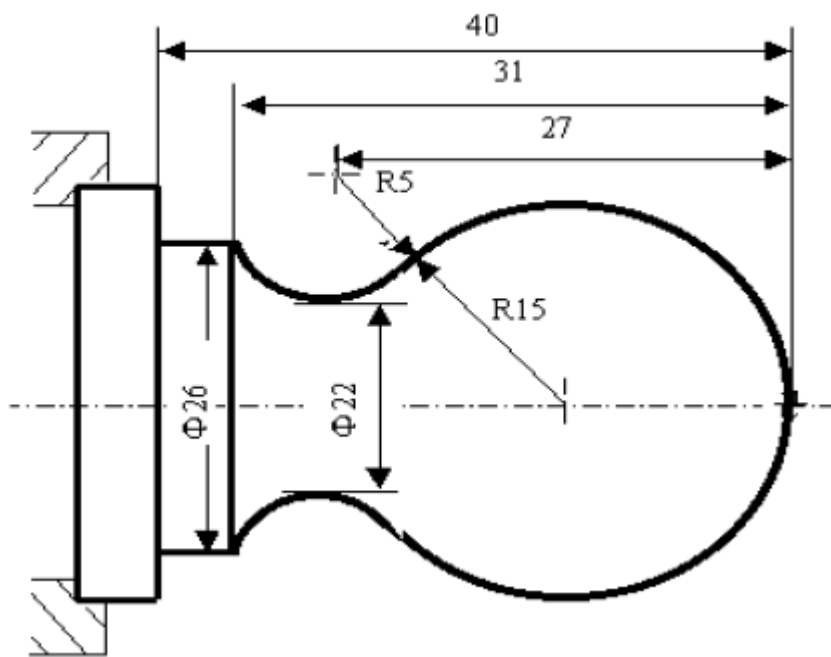
示例：从直径 $\Phi 45.359$ 切削到 $\Phi 63.060$ 的圆弧程序代码。



程序：

```
G02 X63.060 Z-24.071 R19.16 F300 ; 或  
G02 U17.701 W-24.071 R19.16 F300 ; 或  
G02 X63.060 Z-24.071 I17.785 K-7.124 ; 或  
G02 U17.701 W-24.071 I17.785 K-7.124 F300
```

G02/G03代码综合编程实例：



程序：O0001

N001 G0 X40 Z5;	(快速定位)
N002 M03 S200;	(主轴开)
N003 G01 X0 Z0 F900;	(靠近工件)
N005 G03 U24 W-24 R15;	(切削R15 圆弧段)
N006 G02 X26 Z-31 R5;	(切削R5 圆弧段)
N007 G01 Z-40;	(切削Φ26)
N008 X40 Z5;	(返回起点)
N009 M30;	(程序结束)

1.2.4 G04（暂停指令）

指令格式：G04 X（U）_

代码功能：各轴运动停止，不改变当前的G代码模态和保持的数据、状态，延时给定的时间后，再执行下一个程序段。

代码说明：G04为非模态G代码；

G04延时时间由代码字P__、X__或U__指定；

P值取范围为-99999999~99999999（单位：ms）。

X、U代码范围为-9999.999~9999.999（单位：s）

注意事项：

当P、X、U未输入时,表示程序段间准确停。

当P、X、U指定负值时,表示暂停时间为0。

P、X、U在同一程序段，P有效；X、U在同一程序段，X有效。

G04代码执行中,进行进给保持的操作，当前页面下方显示暂停,但G04计时没有停止,当计时时间到时,光标停留到下一段程序。

示例：

G04 X0.5; //暂停0.5秒

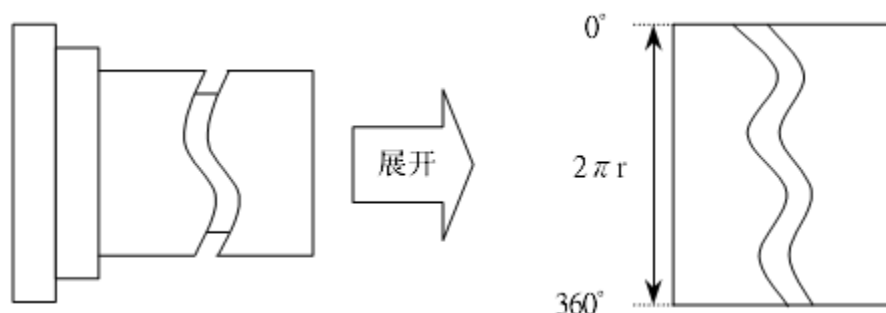
G04 U0.5; //暂停0.5秒

G04 P500; //暂停0.5秒

1.2.5 G07.1（圆柱插补）

指令格式：G07.1 C_

代码功能：用角度指定的回转轴的移动量内部转换为沿外表面的直线轴的距离，以便能同其它轴一起完成直线插补或圆弧插补。在插补完成后，这一距离又转换为回转轴的移动量。圆柱插补用圆柱体的展开面编程（如下图）



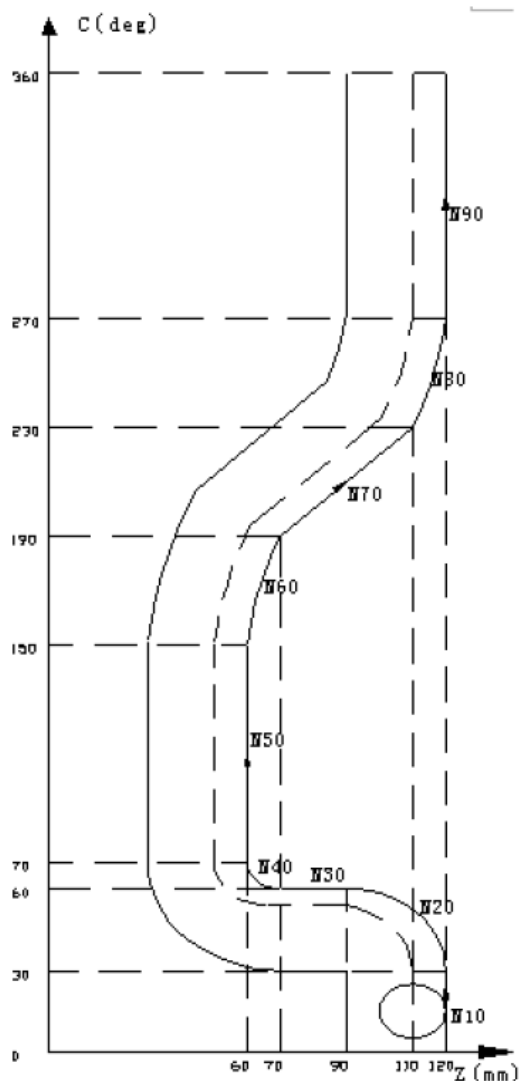
注意事项：

1. 圆柱插补模式中可实现直线G01、圆弧G02、G03（圆弧的半径只能由R指定，单位是mm 或inch）；
2. 圆柱插补模式中不可G00定位操作；
3. 进入圆柱插补方式之前应取消正在进行的刀具半径补偿模式，而在圆柱插补方式内开始并结束刀具补偿；
4. 辅助功能T不能在圆柱插补模式中使用；
5. 圆柱插补进给速度的指定只能是G98分进给；
6. 圆柱插补模式中不可再用G50设定工件坐标系；
7. 圆柱插补中只允许指定当前圆柱的旋转轴及直线轴；

示例:

```

O0071
G18;
G98;
G00 X150 Z105 C0;
G01 X114.598 Z105 F200;
G07.1 C57.299;
G41 G01 Z120;
N10 G01 C30.0;
N20 G03 Z90 C60 R30;
N30 G01 Z70;
N40 G02 Z60 C70 R10;
N50 G01 C150;
N60 G02 Z70 C190 R75;
N70 G01 Z110 C230;
N80 G03 Z120 C270 R75;
N90 G01 C360;
G40 G01 Z105;
G07.1 C0;
M30;
    
```



1.2.6 G12.1（启动极坐标插补）G13.1（取消极坐标插补）

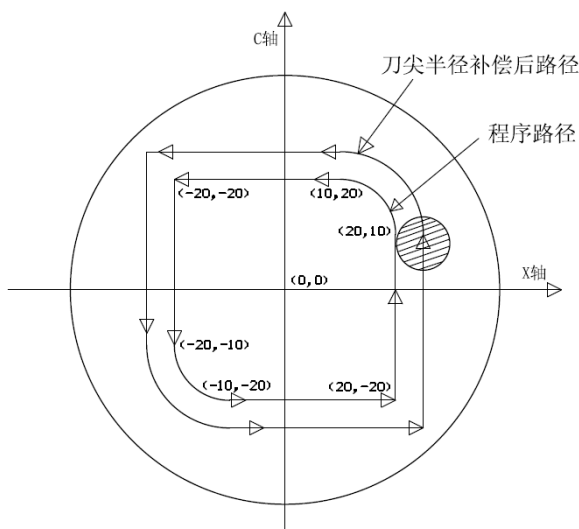
指令格式：G12.1---启动极坐标插补方式（也可以写作G112）

G13.1---取消极坐标插补方式（也可以写作 G113）

代码功能：极坐标插补是一种轮廓控制，它把在笛卡尔坐标系内的编程指令转换为直线轴的移动(刀具的移动)和旋转轴的移动(工件的旋转)。其对车削加工中的正面切口加工和凸轮轴的磨削等有效。

注意事项：

1. G12.1, G13.1 属于21组代码，G12.1, G13.1要单独放在一行。
2. 在G12.1-G13.1 之间不能换刀，换刀和换刀后的定位必须放在G12.1之前。
3. 在C刀补中间或G99状态不能启动极坐标插补，否则报警。
4. 指令了G12.1 时，极坐标插补的刀具位置是从角度0开始的。
5. 系统默认的极坐标插补平面为X轴与C轴平面。



示例：

T0101;

G00 X110 C0 Z0;

//到定位点

G40 G98;

//只能使用G98，不能使用G99

G12.1;

//极坐标插补开始

G42 X55;

//使用直角坐标系X—C平面编写程序，增加一个移动量为0的程序段

G01 X20 F100;

C10;

G03 X10 C20 R10;

G01 X—20;

C—10;

G03 X—10 C—20 R10;

G01 X20;

C0;

G40 X55;

G13.1

//极坐标取消

M30;

1.2.7 G17、G18、G19(平面选择代码)

指令格式:

G17.....XY 平面

G18.....ZX 平面

G19.....YZ 平面

代码功能: 用 G 代码选择圆弧插补的平面或刀具半径补偿的平面

代码说明: G17, G18, G19 为模态G 代码, 在没指令的程序段里, 平面不发生变化。

注意事项:

选择G17、G19平面时要先设定基本轴Y轴有效;

C刀补状态下不能进行平面切换;

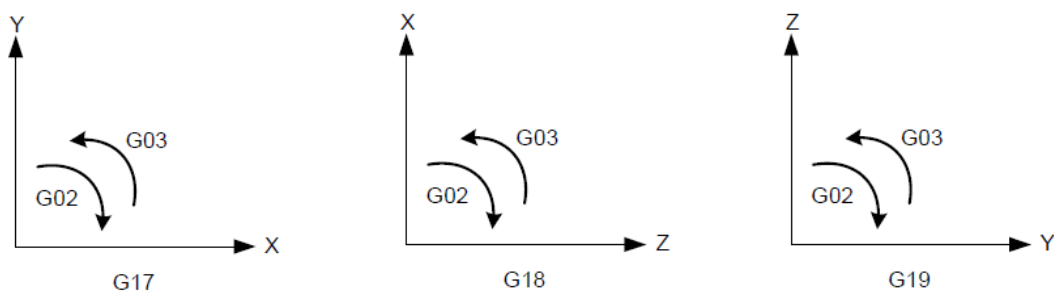
G71~G76, G90, G92, G94只能在G18平面内使用;

平面选择代码可与其他组G代码共段;

移动指令与平面选择无关;

关于直径或半径编程的处理: 由于当前只有一个状态参数010号Bit7选择是直径还是半径编程且只对X轴有效, 因此在使用G2, G3等指令时Z轴与Y轴只能用半径编程, X轴则由参数进行选择; G17和G19平面下的C刀补的刀尖方向为0;

图例:



1.2.8 G90（外圆、内圆车削循环）

指令格式：G90 X(U)___ Z(W)___ F___； （圆柱切削）

G90 X(U)___ Z(W)___ R___ F___；（圆锥切削）

代码功能：从切削点开始，进行径向(X 轴)进刀、轴向(Z 轴或X、Z 轴同时)切削，实现柱面或锥面切削循环。

代码说明：G90 为模态代码；

切削起点：直线插补(切削进给)的起始位置；

切削终点：直线插补(切削进给)的结束位置；

X：切削终点X 轴绝对坐标；

U：切削终点与起点X 轴绝对坐标的差值；

Z：切削终点Z 轴绝对坐标；

W：切削终点与起点Z 轴绝对坐标的差值；

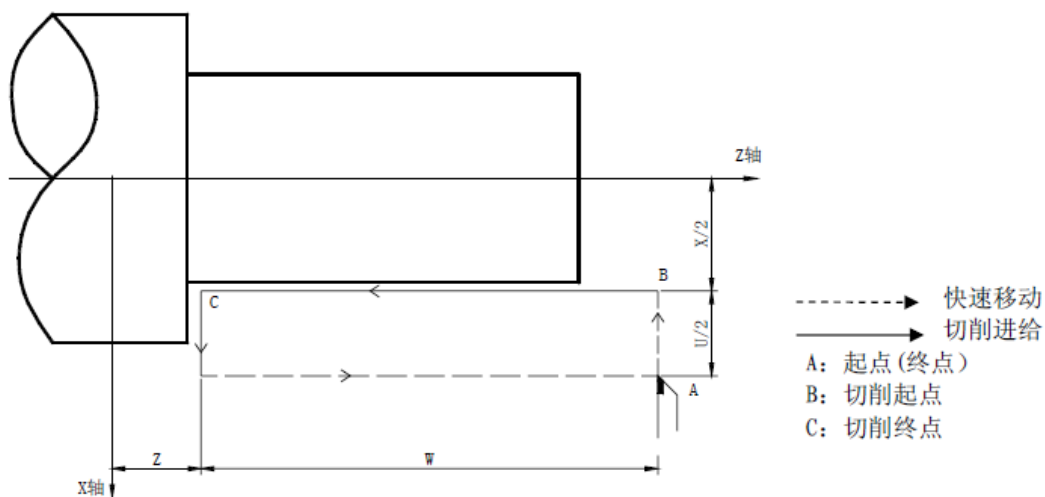
R：切削起点与切削终点X 轴绝对坐标的差值(半径值)，带方向，当R 与U 的符号不一致时，要求 $|R| \leq |U/2|$ ；R=0 或缺省输入时，进行圆柱切削，否则进行圆锥切削。

循环过程：①X 轴从起点快速移动到切削起点B；

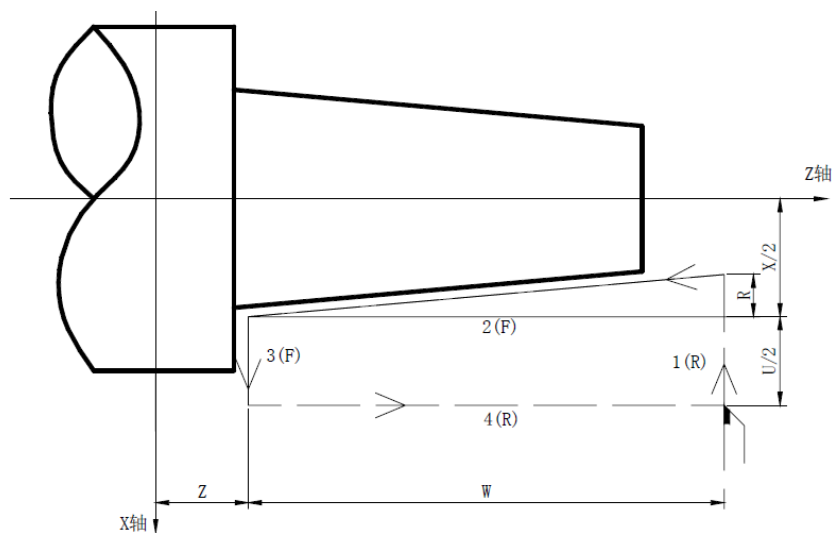
②从切削起点直线插补(切削进给)到切削终点C；

③X 轴以切削进给速度退刀，返回到X轴绝对坐标与起点A相同处；

④Z 轴快速移动返回到起点 A，循环结束。

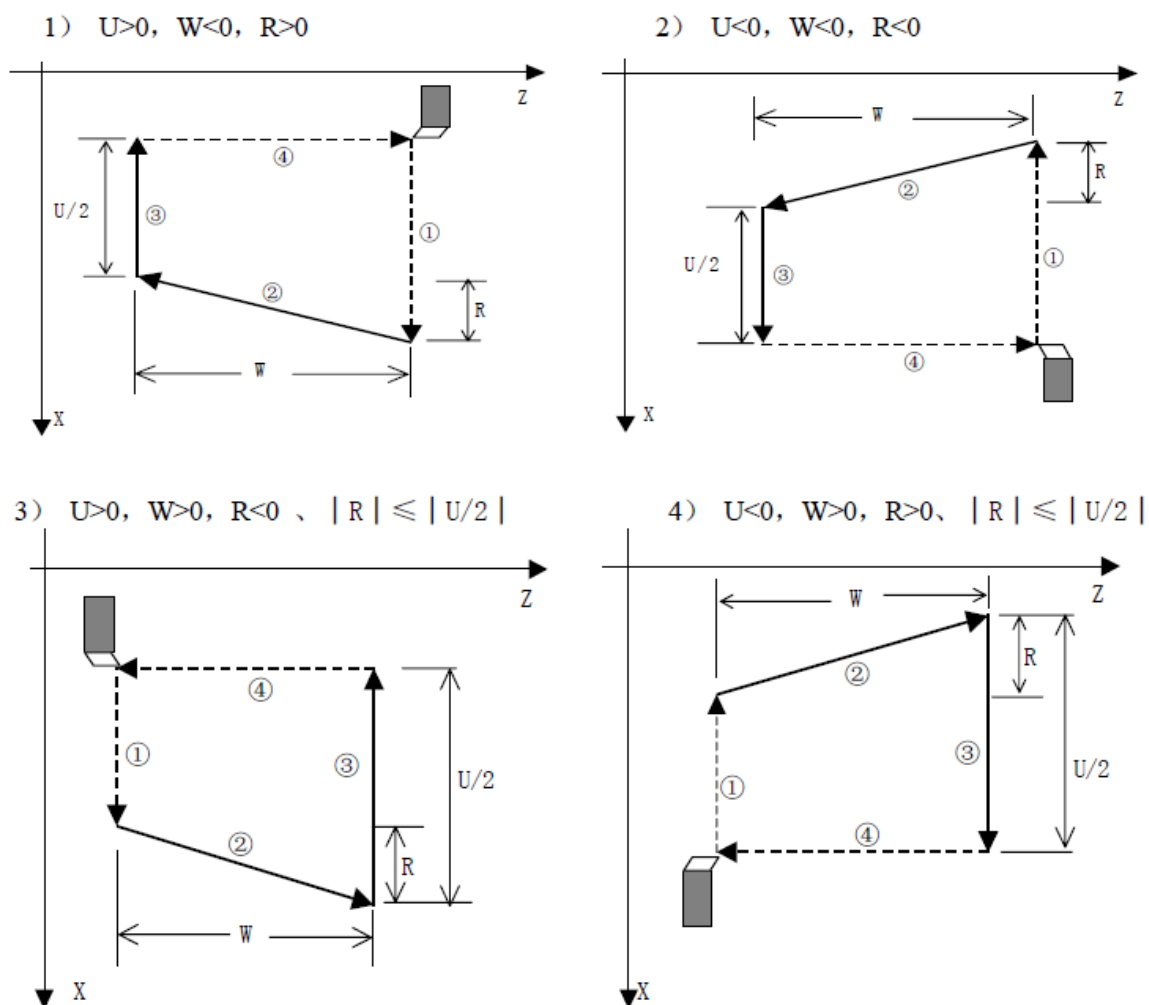


圆柱切削

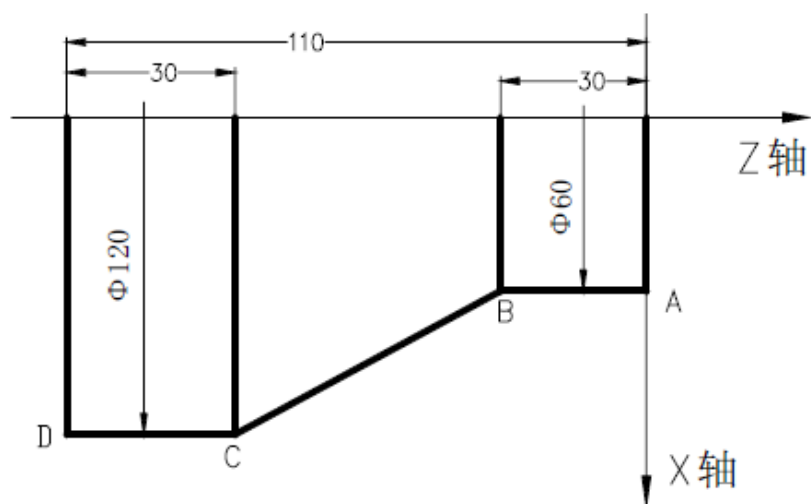


圆锥切削

代码轨迹: U、W、R 反应切削终点与起点的相对位置, U、W、R 在符号不同时组合的刀具轨迹, 如图:



示例：毛坯 $\Phi 125 \times 110$



程序：O0002；

M3 S300 G0 X130 Z3；

G90 X120 Z-110 F200； (A→D, $\Phi 120$ 切削)

X110 Z-30；

X100；

X90；

X80；

X70；

X60；

(A→B, $\Phi 60$ 切削, 分六次进刀循环切削, 每次进刀 10mm)

G0 X120 Z-30；

G90 X120 Z-44 R-7.5 F150；

Z-56 R-15

Z-68 R-22.5

Z-80 R-30

(B→C, 锥度切削, 分四次进刀循环切削)

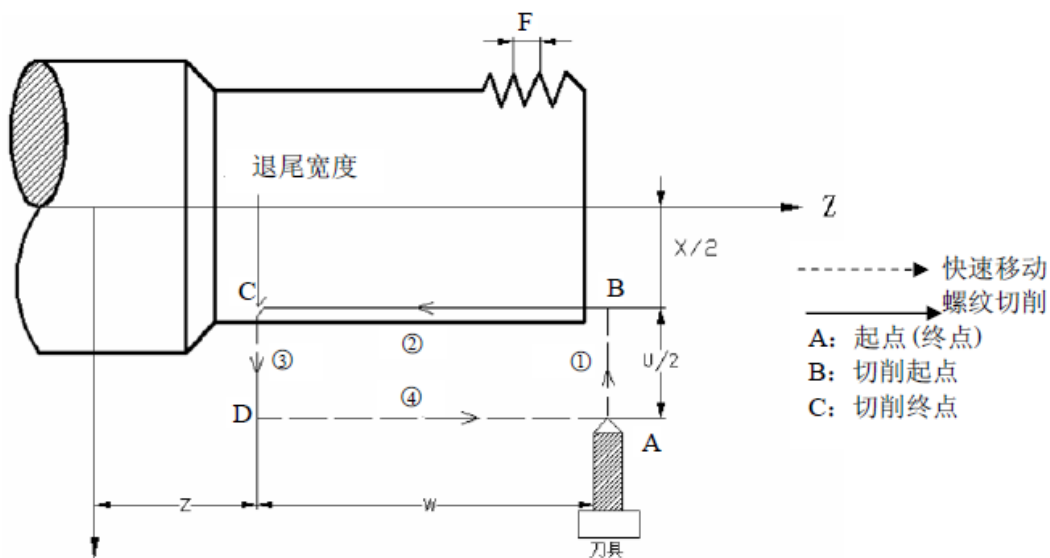
M30；

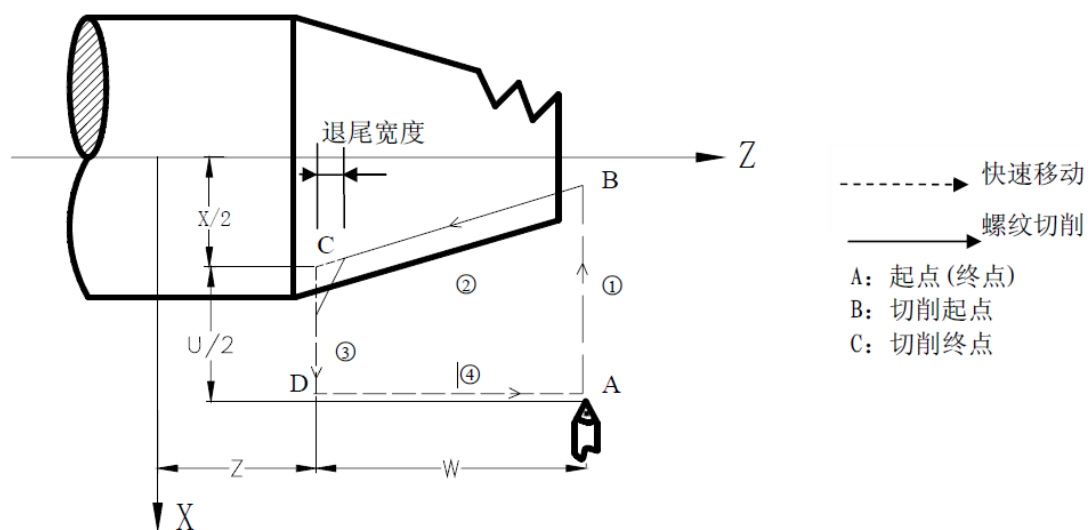
1.2.9 G92(螺纹切削循环)

指令格式: G92 X(U)_ Z(W)_ F_ J_ K_ L ; (公制直螺纹切削循环)
 G92 X(U)_ Z(W)_ I_ J_ K_ L ; (英制直螺纹切削循环)
 G92 X(U)_ Z(W)_ R_ F_ J_ K_ L ; (公制锥螺纹切削循环)
 G92 X(U)_ Z(W)_ R_ I_ J_ K_ L ; (英制锥螺纹切削循环)

代码功能: 从切削起点开始, 进行径向(X轴)进刀、轴向(Z轴或X、Z轴同时)切削, 实现等螺距的直螺纹、锥螺纹切削循环。执行G92代码, 在螺纹加工末端有螺纹退尾过程: 在距离螺纹切削终点固定长度(称为螺纹的退尾长度)处, 在Z轴继续进行螺纹插补的同时, X轴沿退刀方向加速退出, Z轴到达切削终点后, X轴再以快速移动速度退刀。

代码说明: G92为模态G代码;
 切削起点: 螺纹插补的起始位置;
 切削终点: 螺纹插补的结束位置;
 X: 切削终点X轴绝对坐标;
 U: 切削终点与起点X轴绝对坐标的差值;
 Z: 切削终点Z轴绝对坐标;
 W: 切削终点与起点Z轴绝对坐标的差值;
 R: 切削起点与切削终点X轴绝对坐标的差值(半径值), 当R与U的符号不一致时, 要求 $|R| \leq |U/2|$;
 F: 螺纹导程, F指定值执行后保持, 可省略输入;
 I: 螺纹每英寸牙数, I指定值执行后保持, 可省略输入;
 J: 螺纹退尾时在短轴方向的移动量, 取值范围0~99999999×最小输入增量, 单位: mm/inch。不带方向(根据程序起点位置自动确定退尾方向), 模态参数, 如果短轴是X轴, 则该值为半径指定;
 K: 螺纹退尾时在长轴方向的长度, 取值范围0~99999999×最小输入增量, 单位: mm/inch。不带方向, 模态参数, 如长轴是X轴, 该值为半径指定;
 L: 多头螺纹的头数, 该值的范围是: 1~99, 模态参数。(省略L时默认为单头螺纹)





G92 代码可以分多次进刀完成一个螺纹的加工，但不能实现2个连续螺纹的加工，也不能加工端面螺纹。G92代码螺纹螺距的定义与G32 一致，螺距是指主轴转一圈长轴的位移量(X 轴位移量按半径值)。

锥螺纹的螺距是指主轴转一圈长轴的位移量(X轴位移量按半径值)，B点与C点Z轴坐标差的绝对值大于X轴(半径值)坐标差的绝对值时，Z轴为长轴；反之，X轴为长轴。

循环过程：

- ①X 轴从起点快速移动到切削起点B；
- ②从切削起点螺纹插补到切削终点C；
- ③X 轴以快速移动速度退刀(与①方向相反)，返回到X轴绝对坐标与起点A相同处；
- ④Z 轴快速移动返回到起点A，循环结束。

注意事项：

省略J、K时，按数据参数026号设定值退尾；

省略J时，长轴方向按K退尾，短轴方向数据参数026号设定值退尾；

省略K时，按J=K退尾；

J=0或J=0、K=0时，无退尾；

J≠0，K=0时，按K=J退尾；

J=0，K≠0时，无退尾；

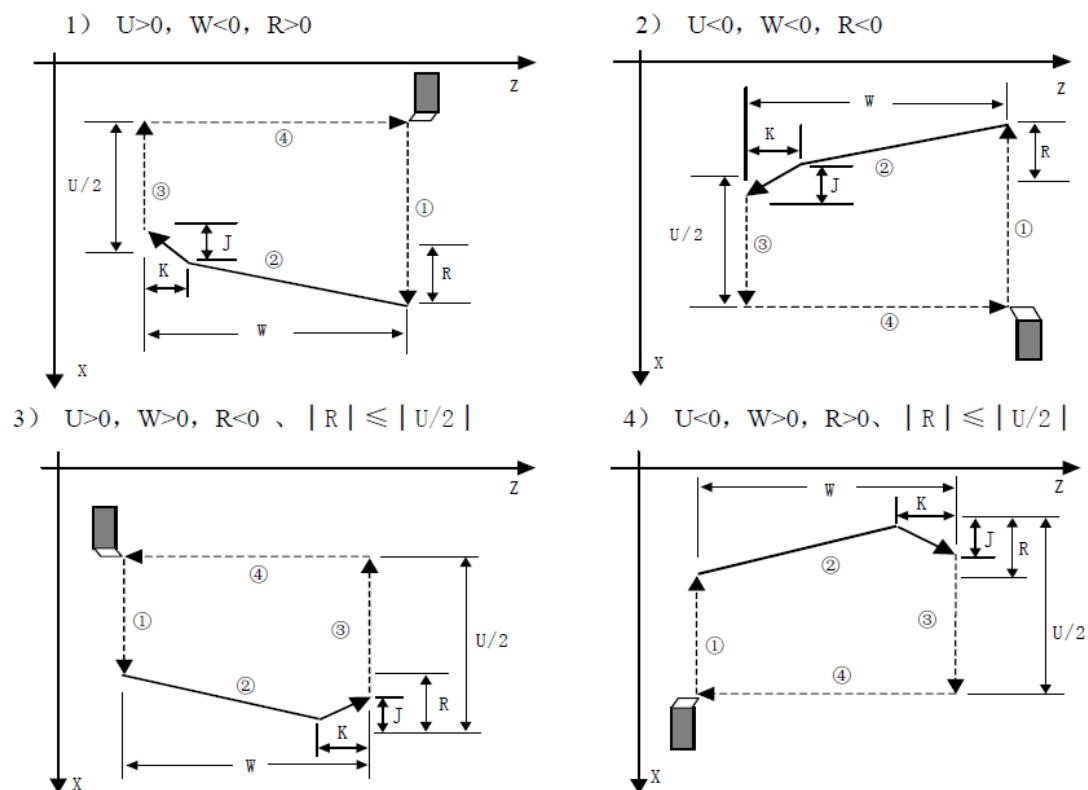
螺纹切削过程中执行进给保持操作后，系统仍进行螺纹切削，螺纹切削完毕，显示“系统暂停”，程序运行暂停；

螺纹切削过程中执行单程式段操作后，在返回起点后(一次螺纹切削循环动作完成)运行停止；

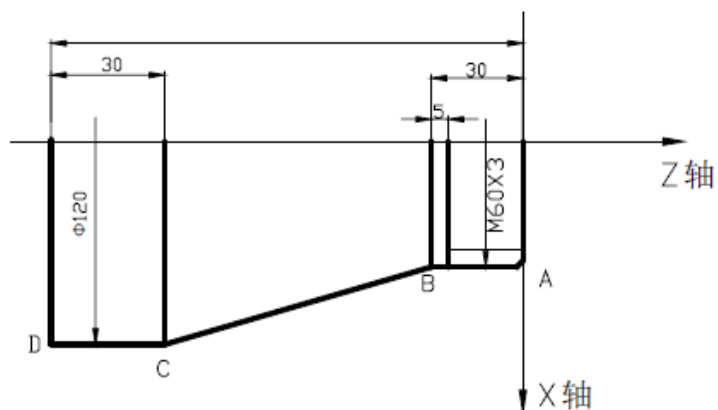
J、K输入负值时，按正值处理；

系统复位、急停或驱动报警时，螺纹切削减速停止。

代码轨迹：U、W、R 反应螺纹切削终点与起点的相对位置，在符号不同时刀具轨迹与退尾方向如图：



示例：



程序：

O0012;

M3 S300 G0 X150 Z50 T0101; (螺纹刀)

G0 X65 Z5; (快速定位)

G92 X58.7 Z-28 F3 J3 K1; (加工螺纹，分4刀切削，第一次进刀1.3mm)

X57.7 ; (第二次进刀1mm)

X57; (第三次进刀0.7mm)

X56.9; (第四次进刀0.1mm)

M30;

1.3.0 G94（端面、锥面切削循环）

指令格式：G94 X(U)___ Z(W)___ F___；（端面切削）

G94 X(U)___ Z(W)___ R___ F___（锥面端面切削）

代码功能：从切削点开始，轴向(Z轴)进刀、径向(X轴或X、Z轴同时)切削，实现端面或锥面切削循环，代码的起点和终点相同。

代码说明：G94 为模态代码；

切削起点：直线插补(切削进给)的起始位置；

切削终点：直线插补(切削进给)的结束位置；

X：切削终点X轴绝对坐标；

U：切削终点与起点X轴绝对坐标的差值；

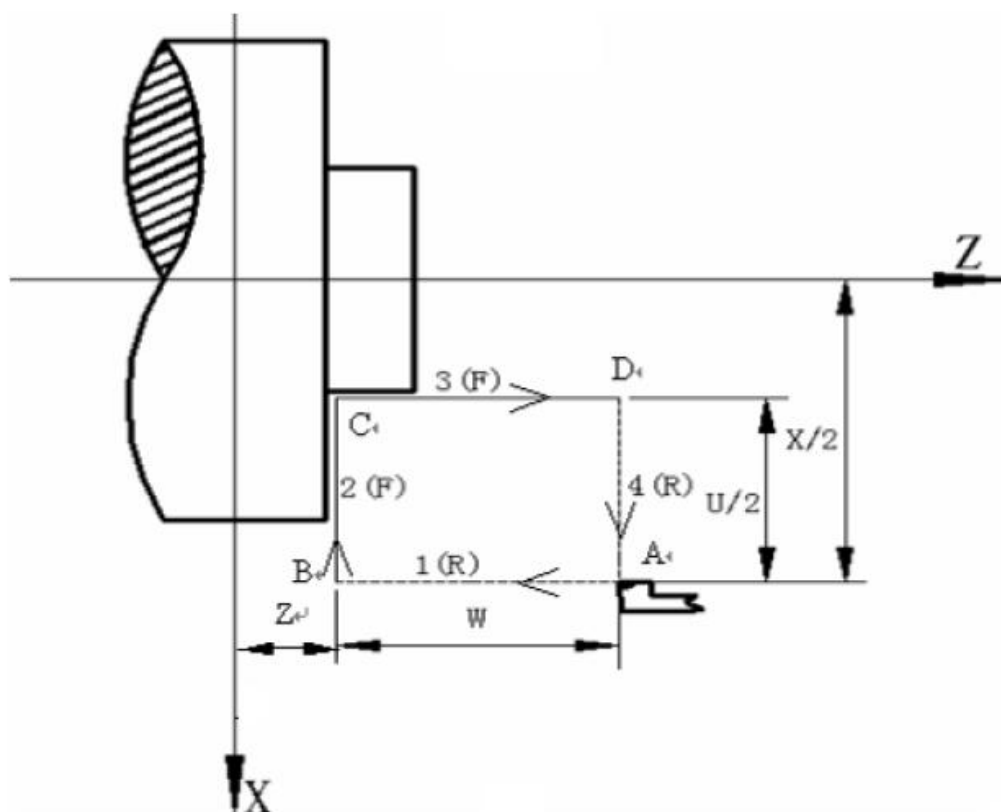
Z：切削终点Z轴绝对坐标；

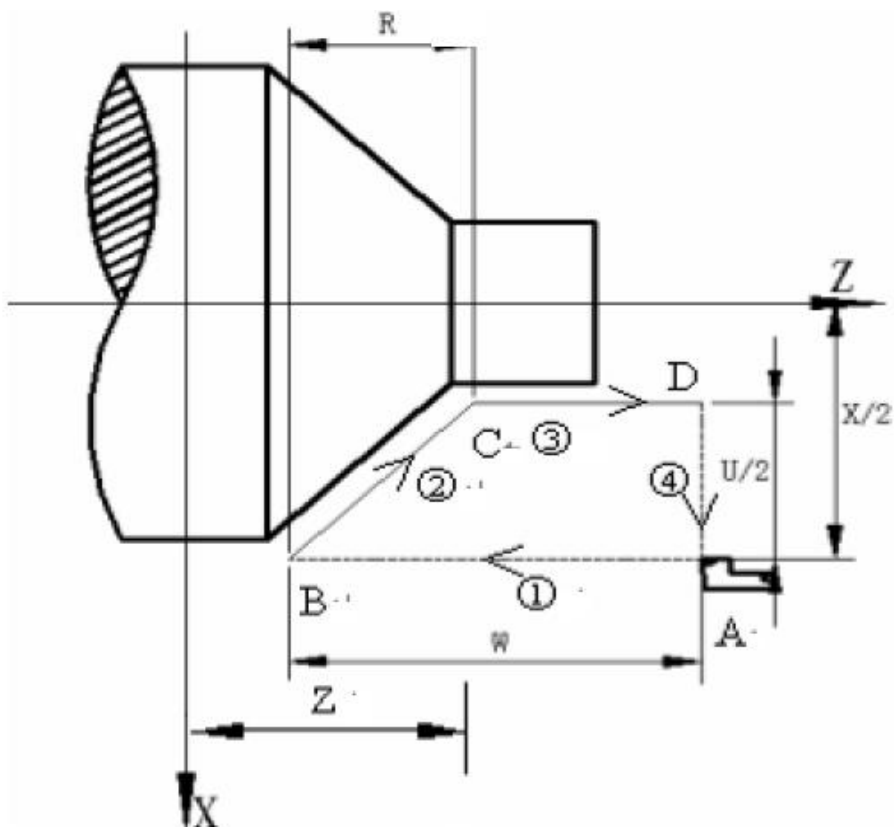
W：切削终点与起点Z轴绝对坐标的差值；

R：切削起点与切削终点Z 轴绝对坐标的差值，当R与W的符号不同时，要求 $|R| \leq |W|$ 。

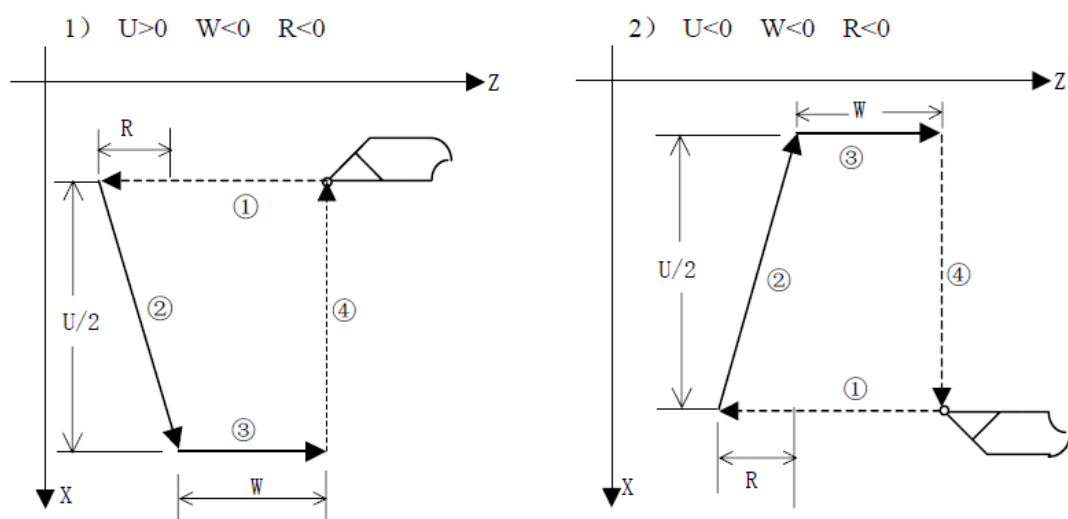
循环过程：

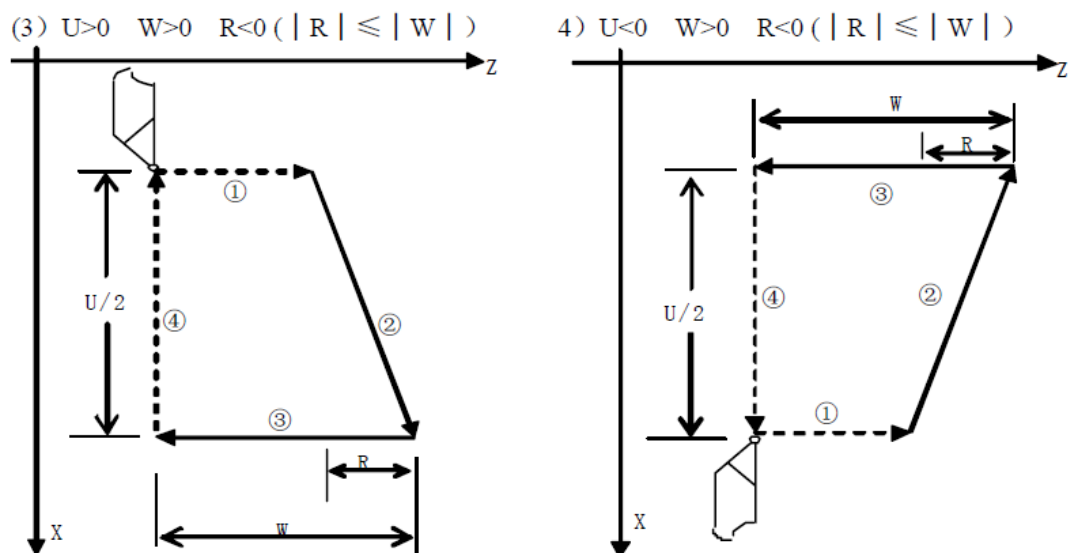
- ② Z 轴从起点快速移动到切削起点B；
- ②从切削起点直线插补(切削进给)到切削终点C；
- ③Z轴以切削进给速度退刀(与①方向相反)，返回到Z轴绝对坐标与起点相同处；
- ④X轴快速移动返回到起点A，循环结束。



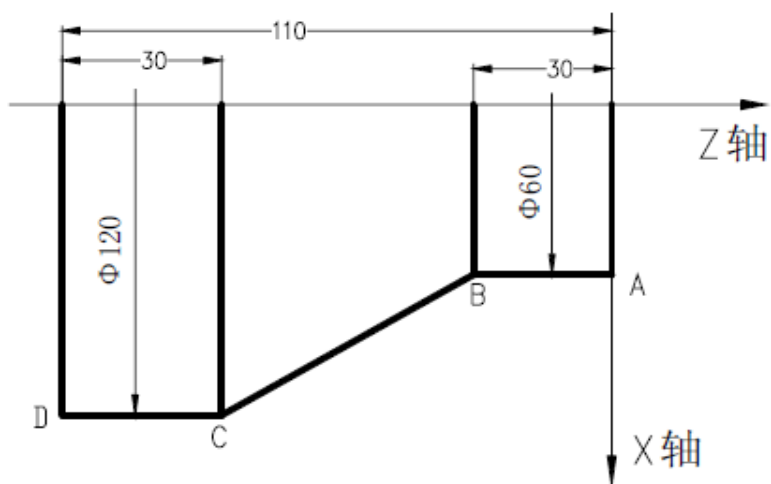


代码轨迹：U、W、R 反应切削终点与起点的相对位置，U、W、R 在符号不同时组合的刀具轨迹，如图：





示例：毛坯 $\Phi 125 \times 112$



程序：O0003；

G00 X130 Z5 M3 S1；

G94 X0 Z0 F200

X120 Z-110 F300；

G00 X120 Z0

G94 X108 Z-30 R-10

X96 R-20

X84 R-30

X72 R-40

X60 R-50；

M30；

端面切削
(外圆 $\Phi 120$ 切削)

(C→B→A, $\Phi 60$ 切削)

1.3.1 G28(返回机床零点)

指令格式: G28 X(U)_ Z(W)_;

代码功能: 从起点开始, 以快速移动速度到达X(U)、Z(W)指定的中间点位置后再回机床零点。

代码说明: G28为非模态G代码;

X、Z: 中间点位置的绝对坐标;

U、W: 中间点位置与起点位置的各轴绝对坐标的差值。

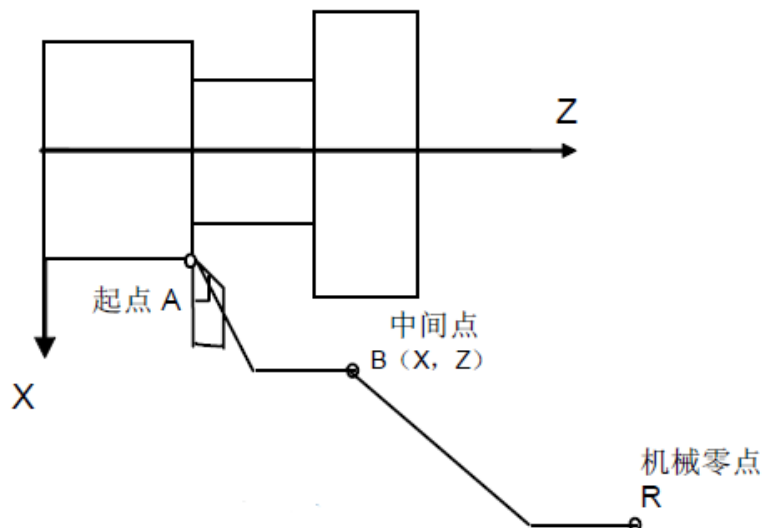
代码地址X(U)、Z(W)可省略一个或全部, 当X或Z后面不输入中间点坐标时, 则不经过中间点直接回零, 详见下表:

指 令	功 能
G28 X(U)_	X轴经过中间点回机床零点, Z轴保持在原位
G28 Z(W)_	Z轴经过中间点回机床零点, X轴保持在原位
G28	保持在原位, 继续执行下一程序段
G28 X(U)_ Z(W)_	X、Z轴同时经过中间点回机床零点
G28 X	X轴不经过中间点直接回机床零点
G28 Z	Z轴不经过中间点直接回机床零点

代码动作过程(如图):

- (1)快速从当前位置定位到中间点位置(A点→B点);
- (2)快速从中间点定位到参考点(B点→R点);
- (3) 若非机床锁住状态, 返回参考点完毕时, 回零灯亮。

图例:



注1: 如果机床未安装零点开关, 不得执行G28代码与返回机床零点的操作。

注2: 手动回机床零点与执行G28代码回机床零点的过程一致, 每次都必须检测减速信号与一转信号;

注3: 从A点→B点及B点→R点过程中, 两轴是以各自独立的快速速度移动的, 因此, 其轨迹并不一定是直线;

注4: 执行G28 代码回机床零点操作后, 系统取消刀具长度补偿;

1.3.2 G31（条件跳转加工）

指令格式：G31 X(U)_ Z(W)_ F_

代码功能：在该代码执行期间，若输入了外部跳转信号，则中断该代码的执行，转而执行下一程序段。该功能可用于工件尺寸的动态测量(如磨床)、对刀测量等。

代码说明：非模态G代码；

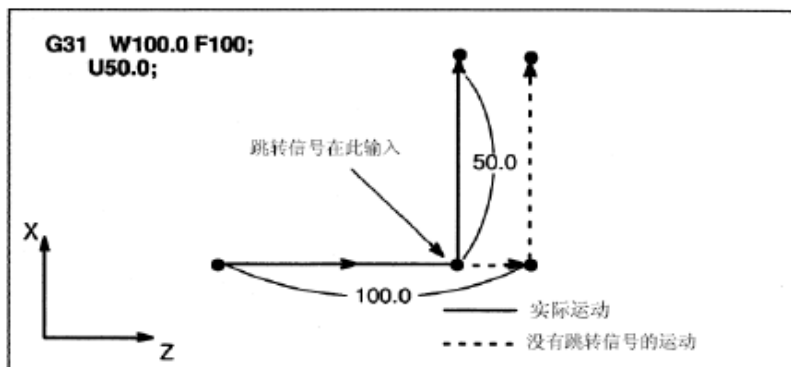
与G01 代码地址格式一致，使用也类似；

使用该代码前需撤销刀尖半径补偿；

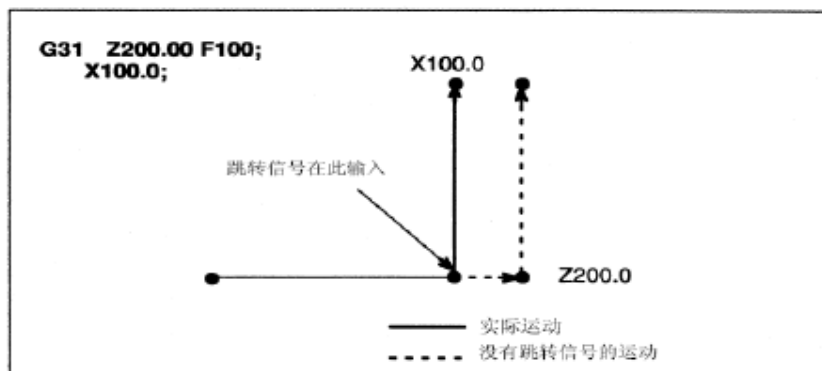
为保证停止位置精度，进给速度不宜设置过大。

跳转发生时后续段的执行：

1: G31的下一个程序段是增量坐标编程, 如下图



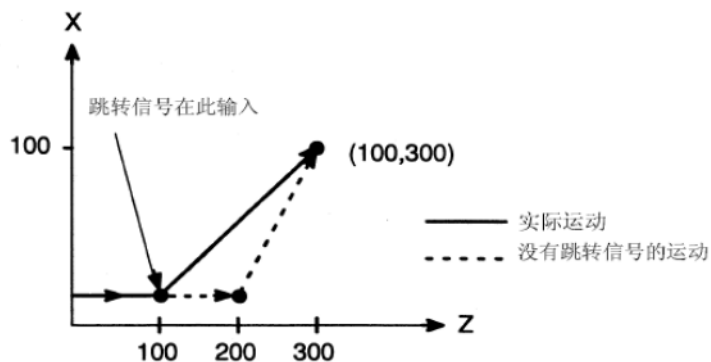
2: G31的下一个程序段是1个轴的绝对坐标编程。



3: G31的下一个程序段是2个轴的绝对坐标编程。

程序：G31 Z200 F100

G01 X100 Z300



1.3.3 G32(螺牙切削)

指令格式: G32 X(U)_ Z(W)_ F(I)_ J_ K_ Q_

代码功能: 刀具的运动轨迹是从起点到终点的一条直线, 如图; 从起点到终点位移量(X轴按半径值)较大的坐标轴称为长轴, 另一个坐标轴称为短轴, 运动过程中主轴每转一圈长轴移动一个导程, 短轴与长轴作直线插补, 刀具切削工件时, 在工件表面形成一条等螺距的螺旋切槽, 实现等螺距螺纹的加工。F、I代码字用于给定螺纹的螺距, 执行G32代码可以加工等螺距的直螺纹、锥螺纹和端面螺纹和连续的多段螺纹加工。

代码说明: G32为模态G代码;

螺纹的导程是指主轴转一圈长轴的位移量(X轴位移量则按半径值);

起点和终点的X坐标值相同(不输入X或U)时, 进行直螺纹切削;

起点和终点的Z坐标值相同(不输入Z或W)时, 进行端面螺纹切削;

起点和终点X、Z坐标值都不相同时, 进行锥螺纹切削。

相关定义:

F: 指定螺纹导程, 为主轴转一圈长轴的移动量, F指定值执行后保持有效, 直至再次执行给定螺纹螺距的F代码字。

I: 指定每英寸螺纹的牙数, 为长轴方向1英寸(25.4 mm)长度上螺纹的牙数, 也可理解为长轴移动1英寸(25.4 mm)时主轴旋转的圈数。I指定值执行后保持有效, 直至再次执行给定螺纹螺距的I代码字。公制输入、英制输入都表示每英寸螺纹的牙数。

J: 螺纹退尾时在短轴方向的移动量(退尾量), 取值范围(-99999999~99999999)×最小输入增量, 单位: mm./inch, 带正负方向; 如果短轴是X轴, 该值为半径指定; J值是模态参数。

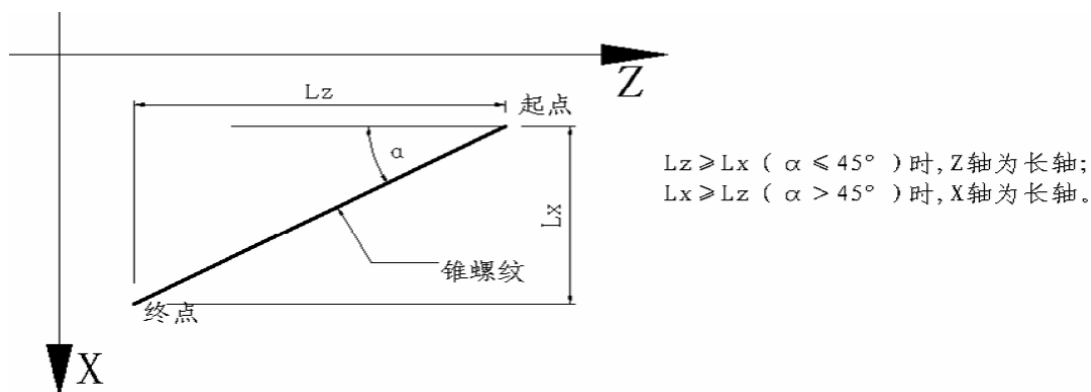
K: 螺纹退尾时在长轴方向的长度, 取值范围0~99999999×最小输入增量, 单位: mm./inch, 如果长轴是X轴, 则该值为半径指定; 不带方向; K值是模态参数。

Q: 起始角, 指主轴一转信号与螺纹切削起点的偏移角度。取值范围0~360000(单位: 0.001度)。Q值是非模态参数, 每次使用都必须指定, 如果不指定就认为是0度。

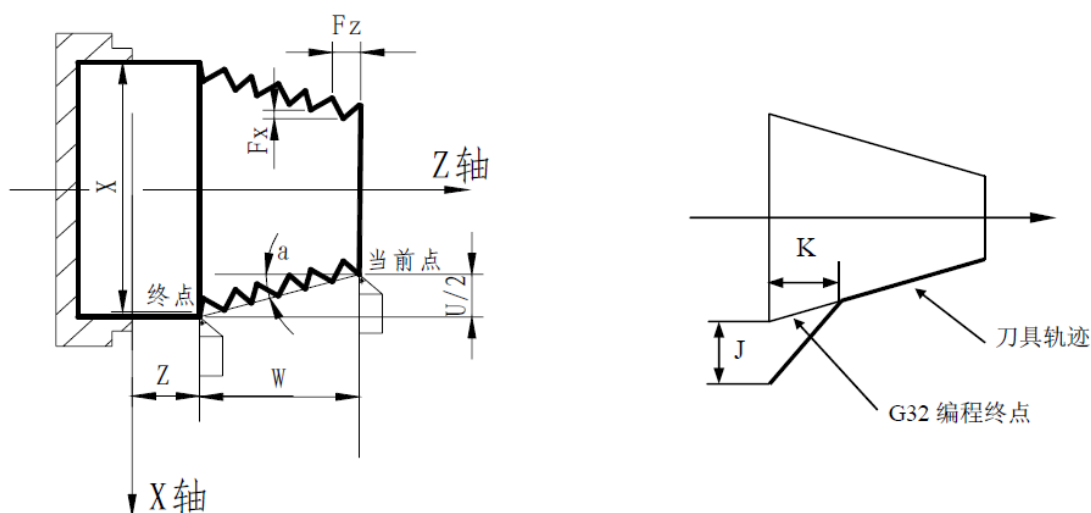
Q使用规则:

- 1、如果不指定Q, 即默认为起始角0度;
- 2、对于连续螺纹切削, 除第一段的Q有效外, 后面螺纹切削段指定的Q无效, 即使定义了Q也被忽略;
- 3、由起始角定义分度形成的多头螺纹总头数不超过65535头。
- 4、Q的单位为0.001度, 若与主轴一转信号偏移180度, 程序中需输入Q180000, 如果输入的为Q180或Q180.0, 均认为是0.18度。

长轴、短轴的判断方法:



G32轨迹图:



注意事项:

J、K是模态代码，连续螺纹切削时下一程序段省略J、K时，按前面的J、K值进行退尾，在执行非螺纹切削代码时取消J、K模态；

省略J或J、K时，无退尾；省略K时，按K=J退尾；

J=0或J=0、K=0时，无退尾；

J≠0，K=0时，按J=K退尾；

J=0，K≠0时，无退尾；

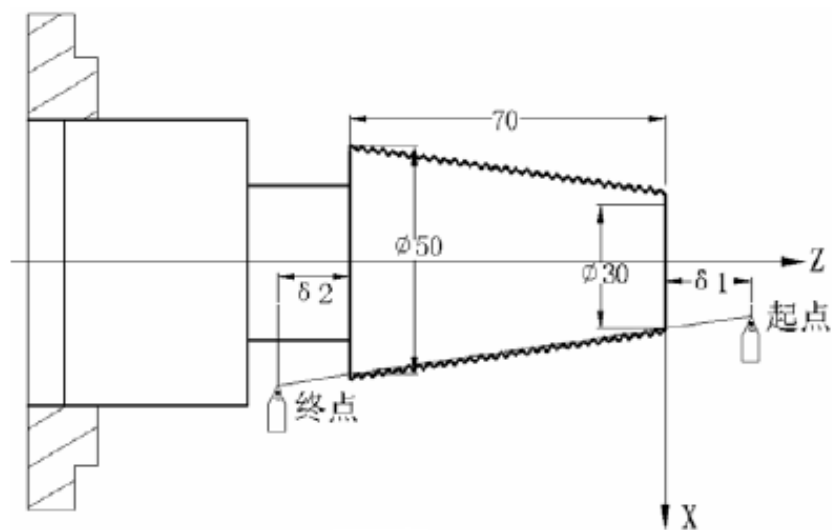
当前程序段为螺纹切削，下一程序段也为螺纹切削，在下一程序段切削开始时不检测主轴位置编码器的一转信号，直接开始螺纹加工，此功能可实现连续螺纹加工。

执行进给保持操作后，系统显示“系统暂停”、螺纹切削不停止，直到当前程序段执行完才停止运动；如为连续螺纹加工则执行完螺纹切削程序段才停止运动，程序运行暂停。

在单段运行，执行完当前程序段停止运动，如为连续螺纹加工则执行完螺纹切削程序段才停止运动。

系统复位、急停或驱动报警时，螺纹切削减速停止。

示例：螺纹螺距：2mm。δ1 = 3mm，δ2 = 2mm，总切深2mm，分两次切入。



程序：O0009；

G00 X28 Z3； (第一次切入1mm)

G32 X50 W-77 F2.0； (锥螺纹第一次切削)

G00 X55； (刀具退出)

W77； (Z 轴回起点)

X27； (第二次再进刀0.5mm)

G32 X49 W-77 F2.0； (锥螺纹第二次切削)

G00 X55； (刀具退出)

W77 ； (Z 轴回起点)

M30；

1.3.4 G33（Z向攻牙循环）

指令格式：G33 Z(W)___ F(I)___ L___；

代码功能：刀具的运动轨迹是从起点到终点，再从终点回到起点。运动过程中主轴每转一圈Z轴移动一个螺距，与丝锥的螺距始终保持一致，在工件内孔形成一条螺旋切槽，可一次切削完成内孔的螺纹加工。

代码说明：G33为模态G代码；

Z(W)：不输入Z或W时，起点和终点的Z坐标值相同，不进行螺纹切削；

F：螺纹导程；

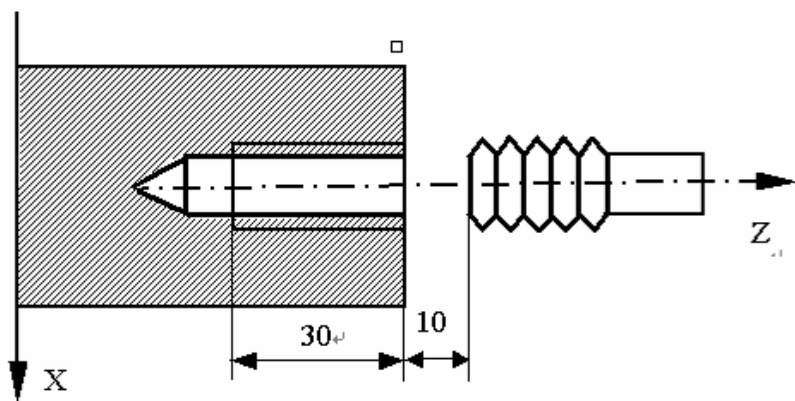
I：每英寸螺纹的牙数；

L：多头螺纹的头数，取值范围1~99，省略L时默认为1头。

循环过程：

- ①Z 轴进刀攻牙(G33代码前必须指定主轴开)；
- ②到达编程指定的Z轴坐标终点后，M05信号输出；
- ③检测主轴完全停止后；
- ④顺时针转信号输出(与原来主轴旋转的方向相反)；
- ⑤Z 轴退刀到起点；
- ⑥M05信号输出，主轴停转；
- ⑦如为多头螺纹，重复①~⑥步骤。

示例：螺纹M10×1.5



程序：

O0011；

G00 Z90 X0 M03； 启动主轴

G33 Z50 F1.5； 攻牙循环

M03； 再启动主轴

G00 X60 Z100； 继续加工

M30；

注1：攻丝前应根据丝锥的旋向来确定主轴旋转方向，攻丝结束后主轴将停止转动，如需继续加工则需要重新启动主轴。

注2：此代码是柔性攻丝，在主轴停止信号有效后，主轴还将有一定的减速时间才停止旋转，此时Z 轴将仍然跟随主轴的转动而进给，直到主轴完全停止，因此实际加工时螺纹的底孔位置应比实际的需要位置稍深一些，具体超出的长度根据攻牙时主轴转速高低和主轴刹车装置而决定。

注3：攻丝切削时Z 轴的移动速度由主轴转速与螺距决定，与切削进给速度倍率无关。

1.3.4 G34（变螺距螺纹切削）

指令格式：G34 X(U)___ Z(W)___ F(I)___ J___ K___ R___ ；

代码功能：刀具的运动轨迹是从X、Z轴起点位置到程序段指定的终点位置的一条直线。从起点到终点位移量(X轴按半径值)较大的坐标轴称为长轴，另一个坐标轴称为短轴，运动过程中主轴每转一圈长轴移动一个导程，并且主轴每转一圈移动的螺距是不断增加指定的值或减少指定的值，在工件表面形成一条变螺距的螺旋切槽，实现变螺距螺纹的加工。切削时，可以设定退刀量。

F、I代码字分别用于指定螺纹的螺距，执行G34代码可以加工公制或英制变螺距的直螺纹、锥螺纹和端面螺纹。

代码说明：G34为模态G代码；

X(U)、Z(W)、J、K 的意义与G32一致；

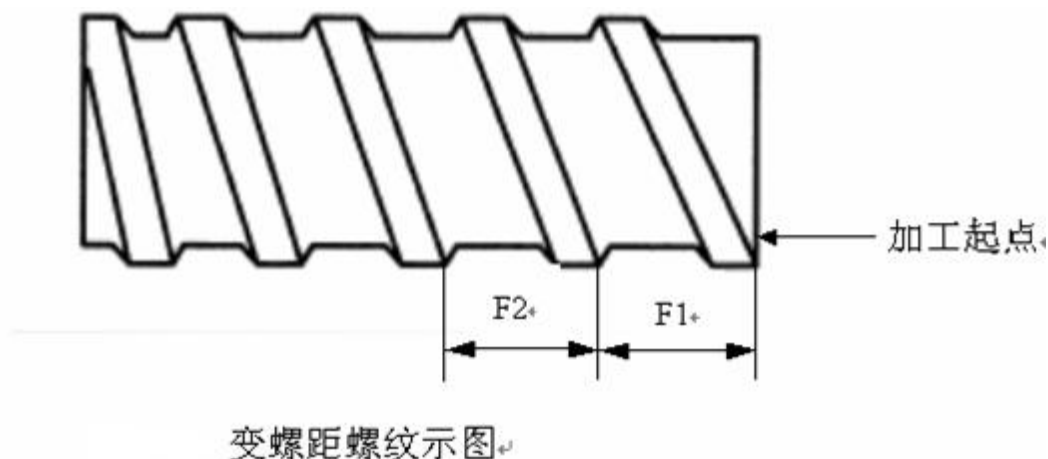
F：指定螺纹的导程；

I：指定每英寸螺纹的牙数；

R：主轴每转螺距的增量值或减量值， $R=F_2-F_1$ ，R带有方向； $F_1>F_2$ 时，R为负值时螺距递减； $F_1<F_2$ 时，R为正值时螺距递增(如图3)；

R 值的范围： $\pm 0.001 \sim \pm 500.000$ 毫米/每螺距(公制螺纹)； $\pm 0.060 \sim \pm 2540$ 牙/每英寸(英制螺纹)。

当R值超过上述范围值和因R的增加/减小使螺距超过允许值或螺距出现负值时产生报警。



示例：

O0001();	
T03;	使用3号刀具
M03 S1000;	主轴正转1000转
M08;	打开冷却液
G00 X0 Z0;	快速定位至切削起始点
G34 Z-50.0 F1.0 R0.2;	以每转螺距增加0.2mm进行螺纹切削
M09;	关闭冷却液
M05;	主轴停止
M30;	程序结束

1.3.5 G36、G37(X、Z轴自动刀具补偿测量)

指令格式: G36 X__;

G37 Z__;

代码功能: 当执行该代码使刀具移至测量位置时, CNC 系统自动测量当前实际坐标值和代码的坐标值之间的差并将其作为刀具的偏移值。该功能常用于自动对刀。

代码说明: X轴绝对坐标(只用于G36), Z轴绝对坐标(只用于G37);

非模态G 代码;

使用该代码前需撤销刀尖半径补偿;

只能使用绝对编程;

使用该代码前必须先定义工件坐标系;

使用该代码前须指定刀号和刀补号;

与G36、G37 自动刀具偏移代码有关的信号:

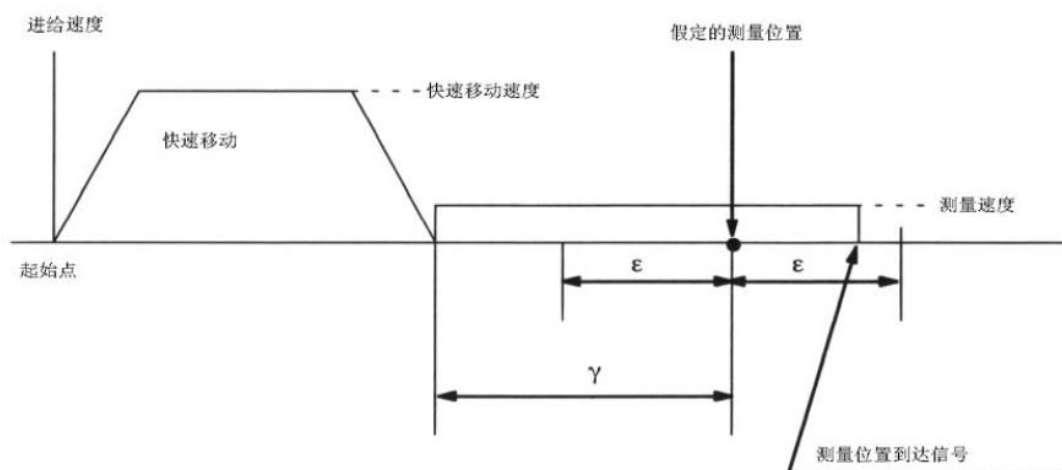
测量位置到达信号:

XAE———对应G36代码

ZAE———对应G37代码

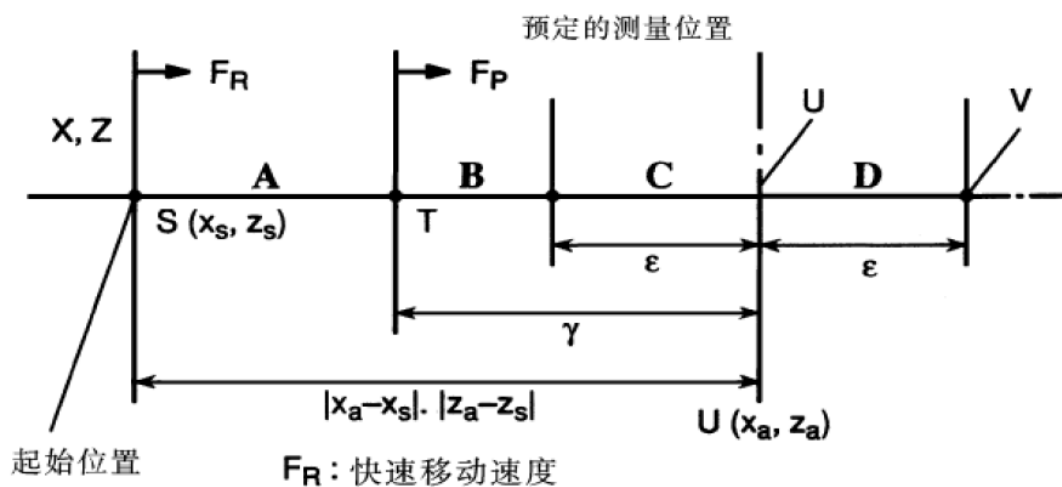
类型: 输入信号

功能: 如果由程序代码的测量位置不同于刀具实际已经到达的测量位置(即测量位置到达信号变为“1”的瞬间位置)的话, 则坐标值的差值被加到当前刀具补偿值中, 以更新补偿值。在执行程序段G36 X__(或G37 Z__)时, 刀具首先快速移动到代码的测量位置, 刀具减速并暂时地停在测量位置之前的距离 γ 处, 然后, 刀具以数据参数084号预置的速度运动到测量位置。如果对应于G代码的测量位置到达信号变为“1”, 在刀具处于测量位置的 $\pm\epsilon$ 的范围内时, CNC 更新偏置补偿值, 并结束这个程序段的运动命令。如果测量位置到达信号不变为“1”, 在刀具已经冲过测量位置距离 ϵ 之后, CNC产生报警, 并且结束该程序段的运行命令, 不更新偏置补偿值。



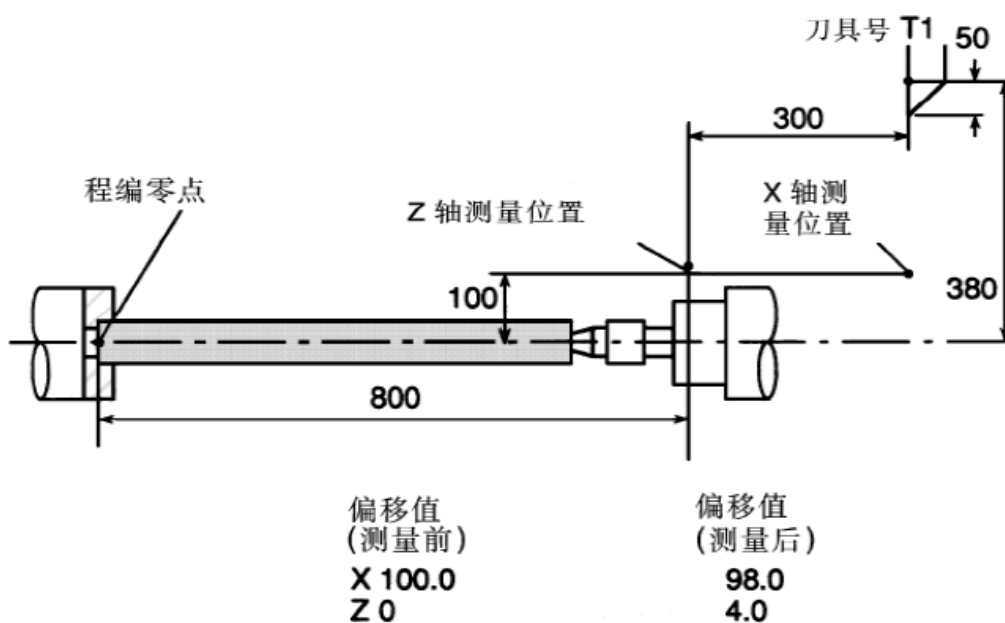
G36、G37 自动刀具偏移代码的使用

从起始位置向着由G36 或G37 中Xa 或Za 指定的测量位置移动时刀具以快速移动速度进给而跨过A区然后刀具停在T 点($Xa-\gamma_x$ 或 $Za-\gamma_z$) 接着以数据参数084号设定的测量进给速度运动而跨过B, 进入C和D区。如果在越过B区时终点的测量位置到达信号置1 则产生报警; 如果在到达V 点以后信号未置1, 刀具停在V 点并产生报警。



示例:

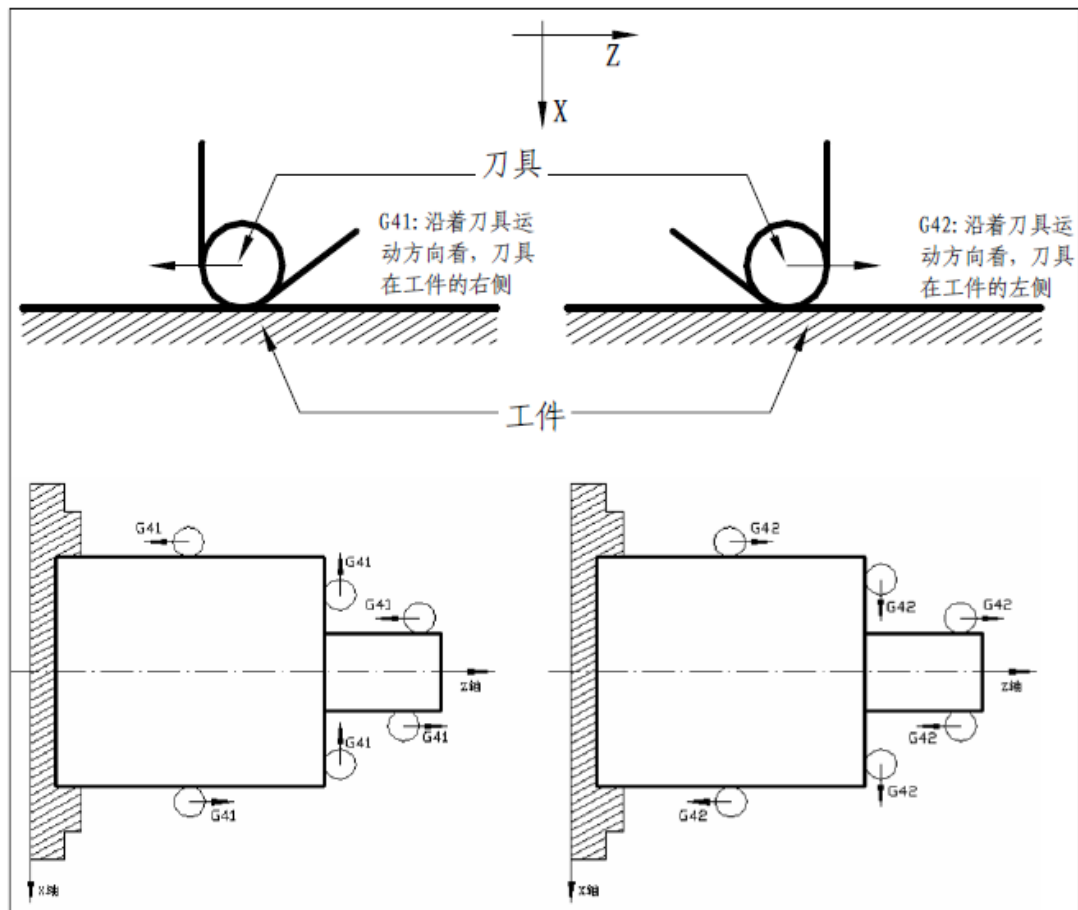
G50 X760 Z1100;	必须建立了工件坐标系
T0101;	定义1号刀, 并使用01号刀补
G36 X200;	向X方向对刀点移动(对刀点的X坐标为200)
T0101;	重新获得X刀补
G00 X204;	倒退一点点
G37 Z800;	向Z方向对刀点移动(对刀点的Z坐标为800)
T0101;	重新获得Z刀补, 对刀完成
M30;	



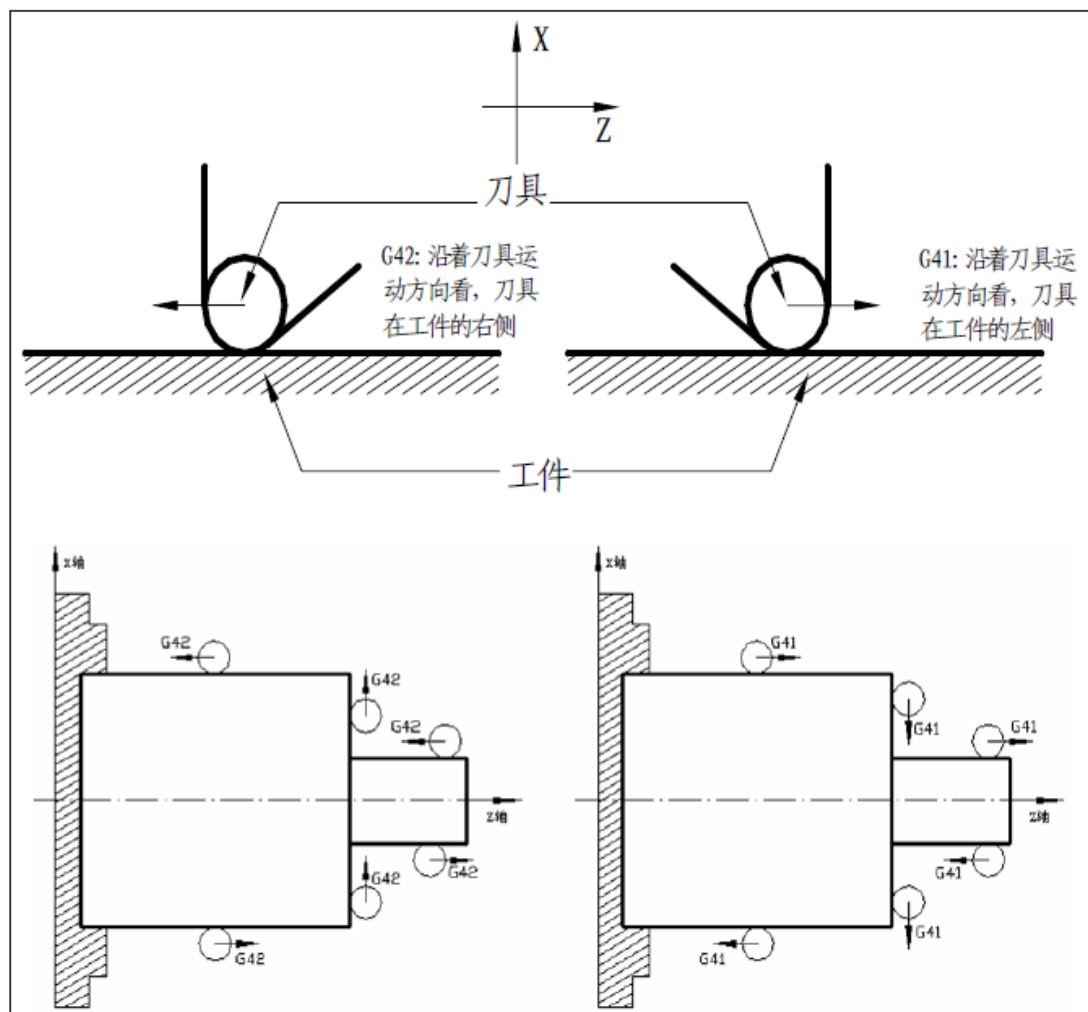
1.3.6 G40、G41、G42(刀具半径补偿取消/左补偿/右补偿)

指令格式: G41 X(U)_Z(W)_
 G42 X(U)_Z(W)_
 G40 补偿取消

补偿方向: 应用刀尖半径补偿, 必须根据刀尖与工件的相对位置来确定补偿的方向。



前刀座坐标系补偿方向



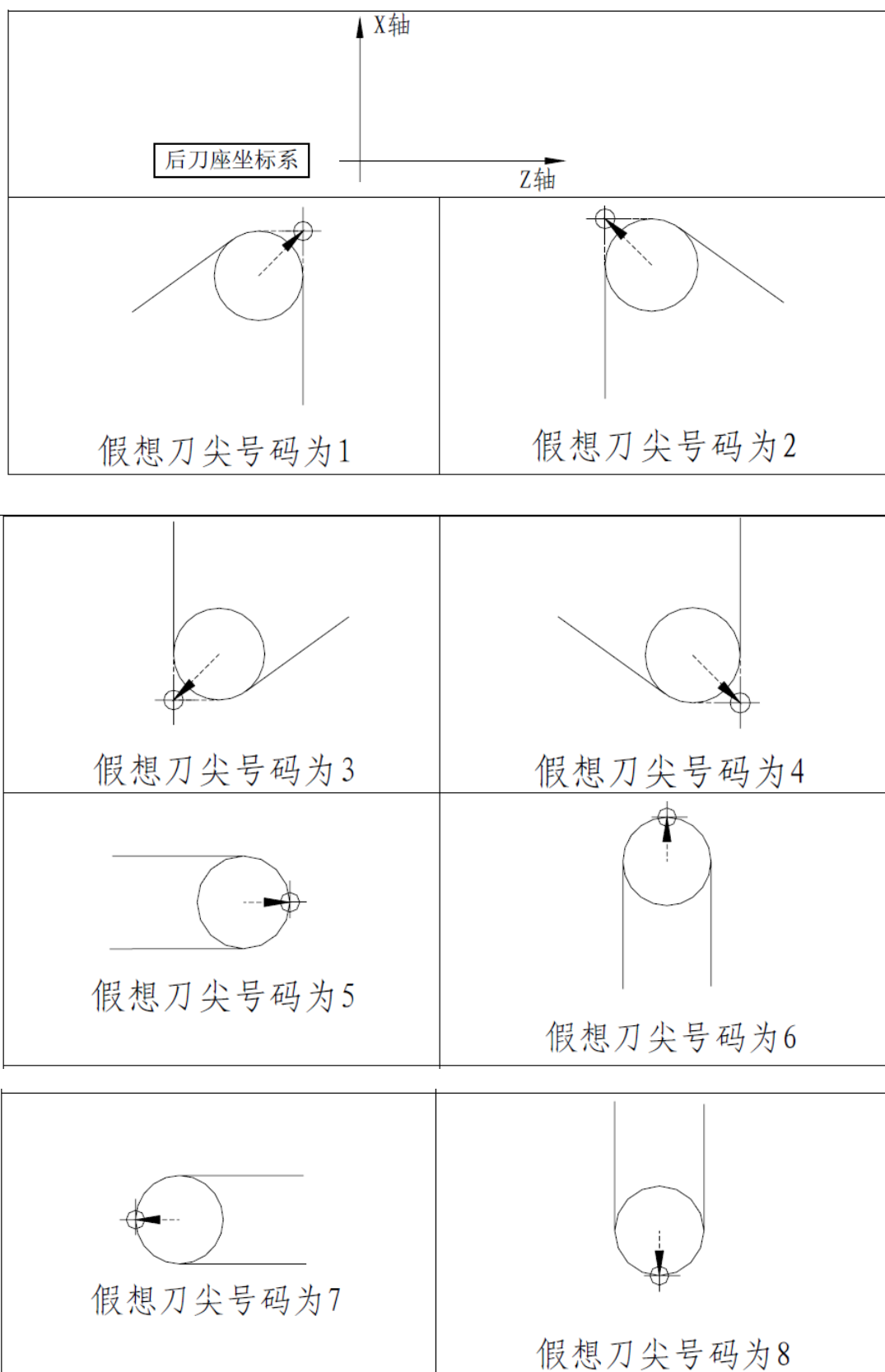
后刀座坐标系补偿方向

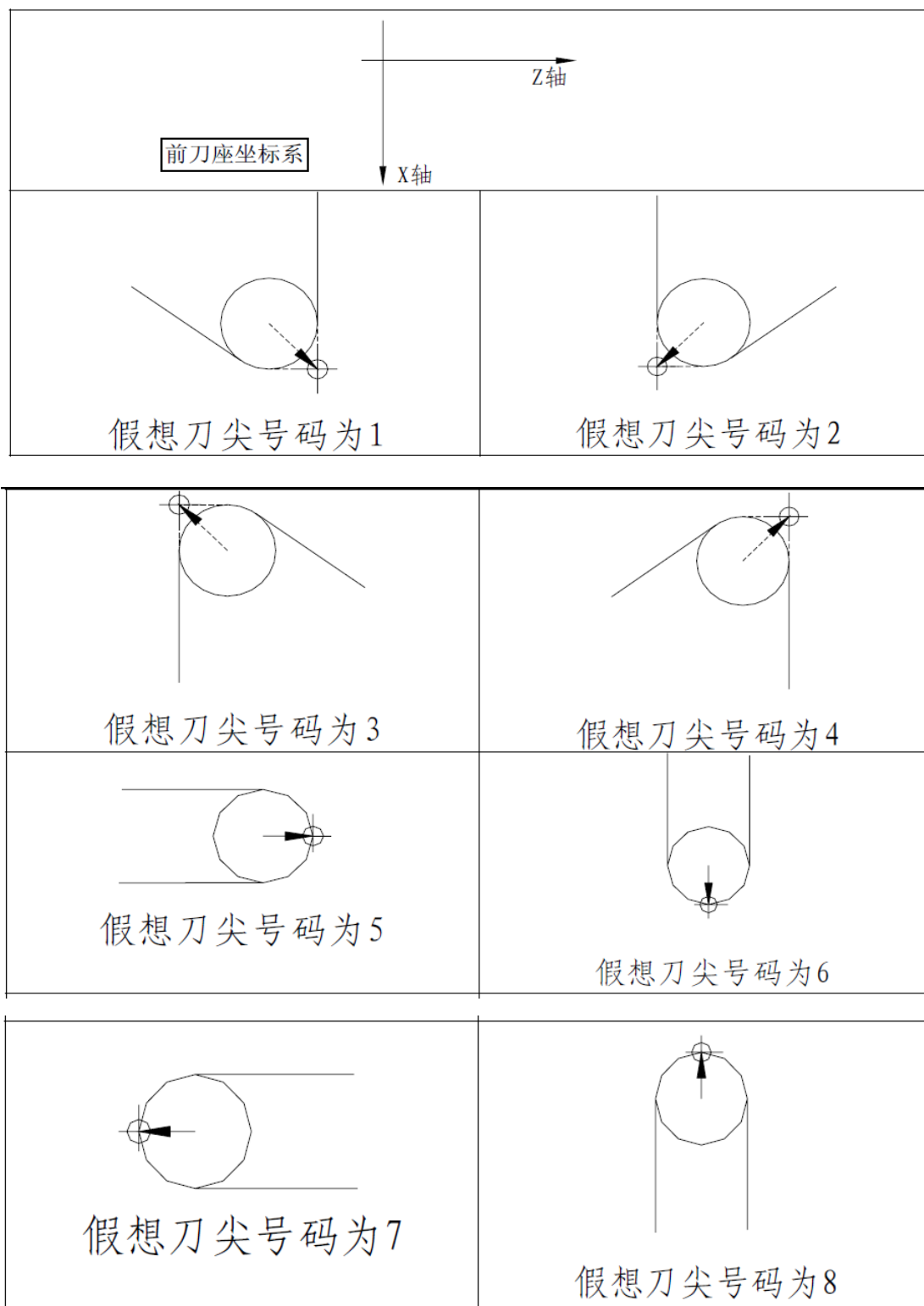
代码功能: 在车削刀具的尖端上磨成小而圆的以尖端以增加刀尖的强度、延长刀具寿命、帮助散热及产生光滑的加工表面。在刀具圆弧所造成误差, 就由G41、G42来完成, 自动将刀尖半径误差精准算出来补正。

注意事项:

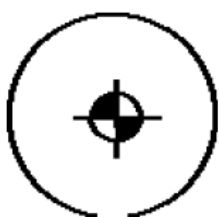
1. 刀尖半径补偿中, 处理 2 个或两个以上无移动代码的程序段时 (如辅助功能, 暂停等), 刀尖中心会移到前一程序段的终点并垂直于前一程序段程序路径的位置。
2. 在录入方式 (MDI) 下不能执行刀补C建立, 也不能执行刀补C撤消。
3. 刀尖半径R值不能输入负值, 否则运行轨迹出错。
4. 尖半径补偿的建立与撤消只能用 G00 或G01 代码, 不能是圆弧代码 (G02 或G03)。如果指定, 会产生报警。
5. 程序结束前必须指定 G40 取消偏置模式。否则, 再次执行时刀具轨迹偏离一个刀尖半径值。
7. 在主程序和子程序中使用刀尖半径补偿, 在调用子程序前 (即执行M98 前), CNC 必须在补偿取消模式, 在子程序中再次建立刀补C。
8. G71、G72、G73、G74、G75、G76 代码不执行刀尖半径补偿, 暂时撤消补偿模式。
9. G90 、G94 代码在执行刀尖半径补偿, 无论是G41 还是G42 都一样偏移一个刀尖半径 (按假想刀尖0号) 进行切削。

假想刀尖号码设定：





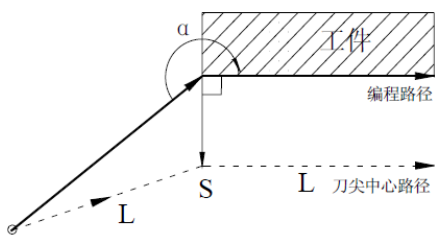
假想刀尖0、9号：



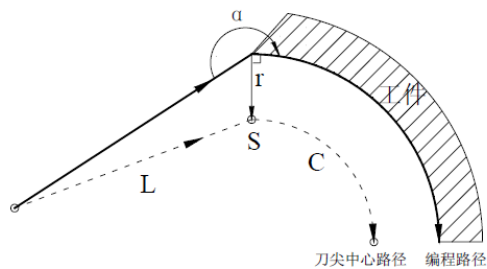
补偿开始起刀时的刀具移动:

沿着拐角的内侧移动 ($\alpha \geq 180^\circ$):

1) 直线→直线

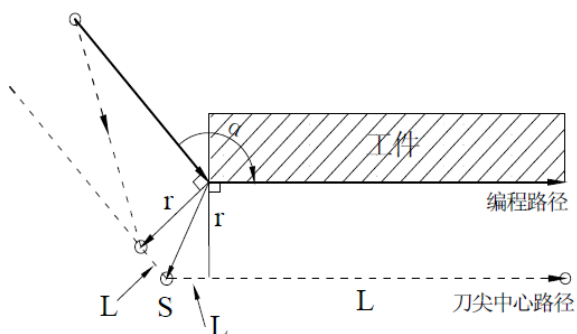


2) 直线→圆弧

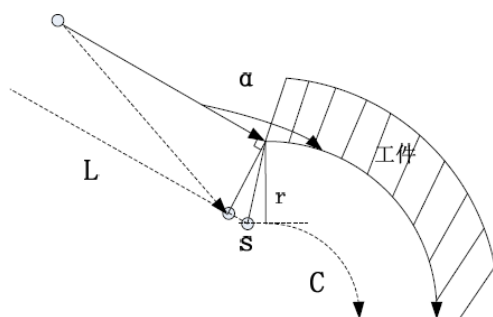


沿着拐角为钝角的外侧移动 ($180^\circ > \alpha \geq 90^\circ$):

1) 直线→直线

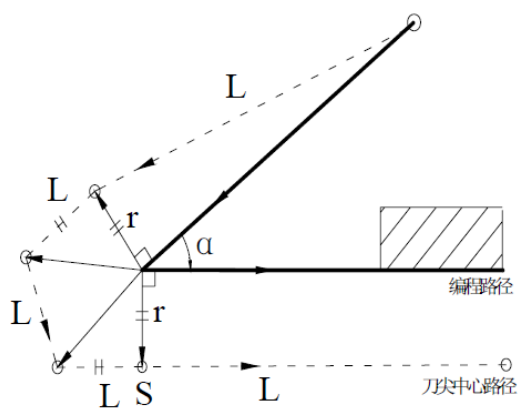


2) 直线→圆弧

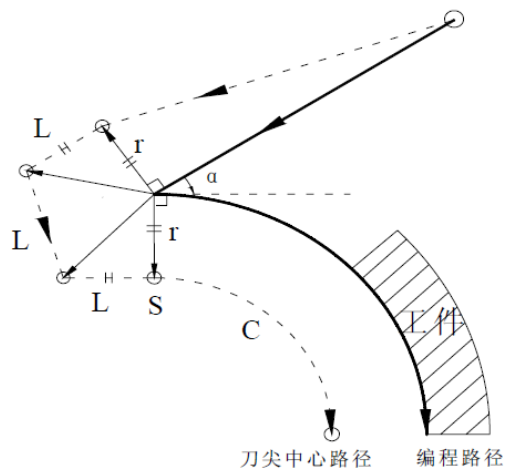


沿着拐角为锐角的外侧移动 ($\alpha < 90^\circ$):

1) 直线→直线



2) 直线→圆弧

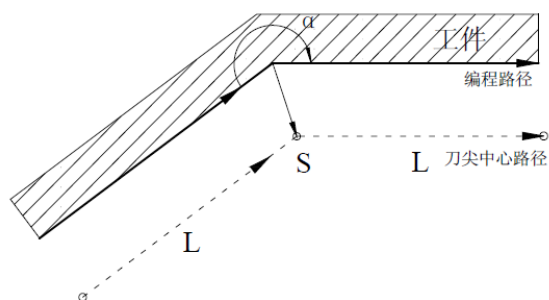


补偿运行中的刀具移动：

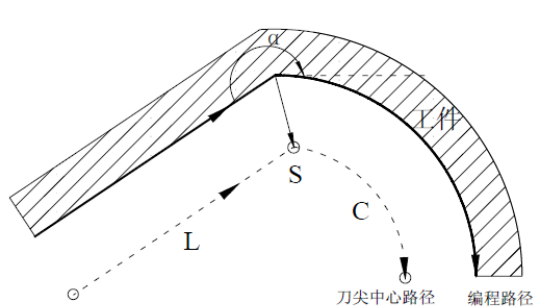
在建立刀尖半径补偿后、取消刀尖半径补偿前称为偏置方式。

沿着拐角的内侧移动 ($\alpha \geq 180^\circ$)：

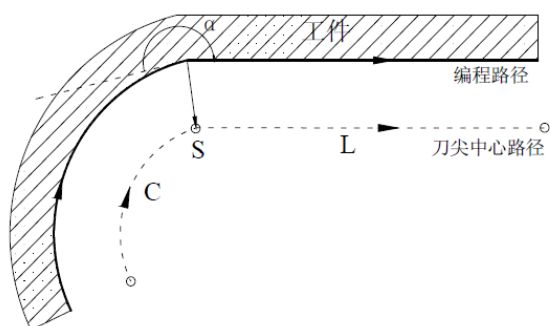
1) 直线→直线



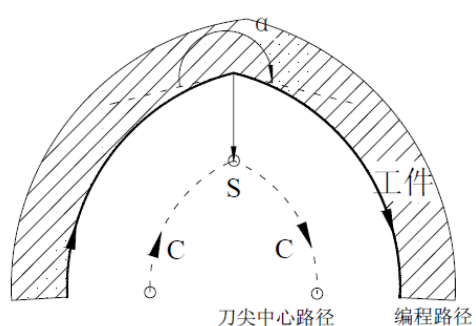
2) 直线→圆弧



3) 圆弧→直线

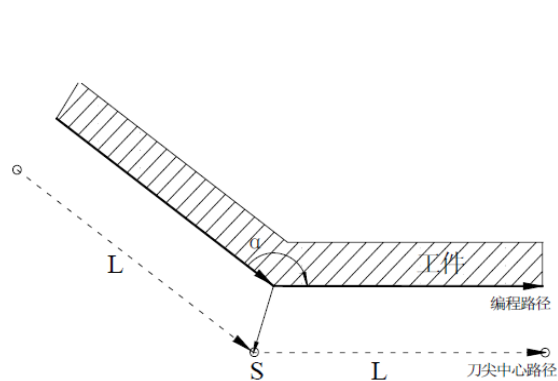


4) 圆弧→圆弧

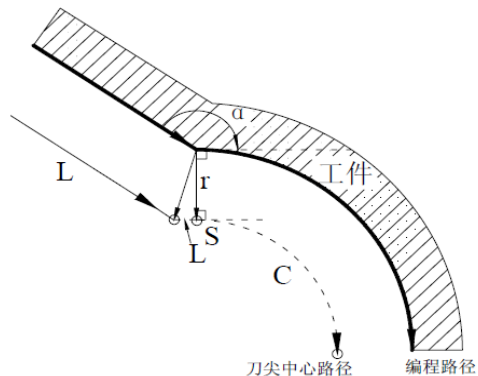


沿着拐角为钝角的外侧移动 ($180^\circ > \alpha \geq 90^\circ$)：

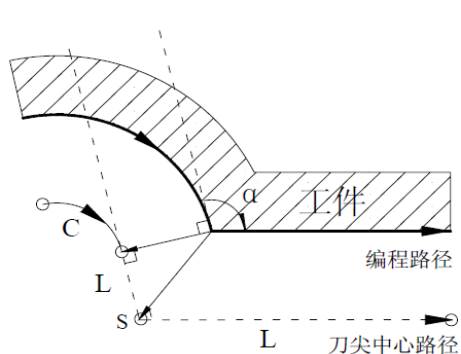
1) 直线→直线



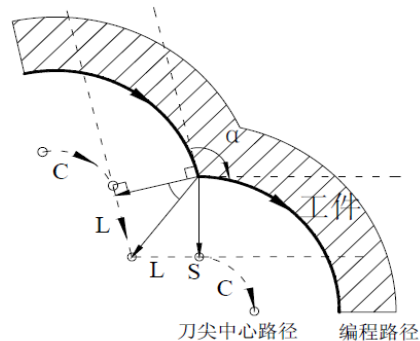
2) 直线→圆弧



3) 圆弧→直线

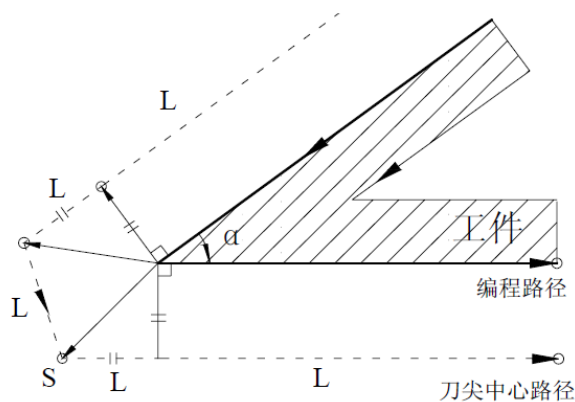


4) 圆弧→圆弧

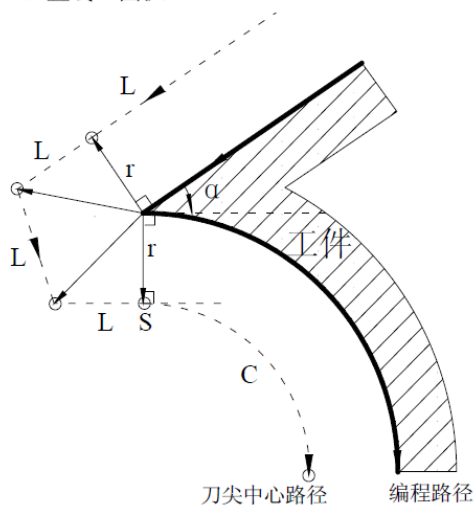


沿着拐角为锐角的外侧移动 ($\alpha < 90^\circ$) :

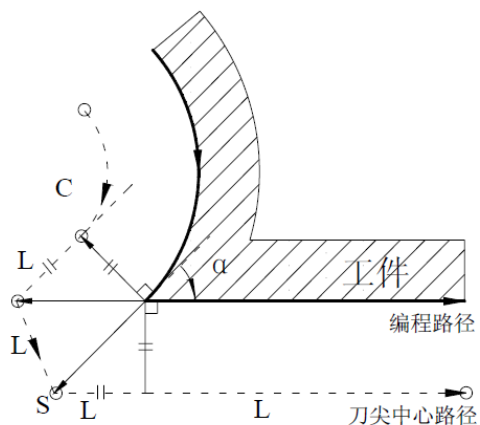
1) 直线→直线



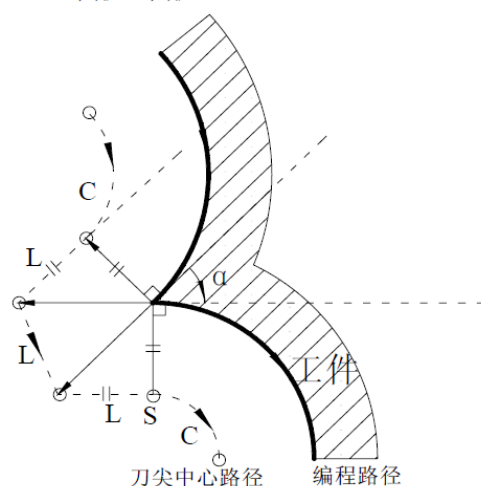
2) 直线→圆弧



3) 圆弧→直线



4) 圆弧→圆弧



补偿偏置取消方式中的刀具移动:

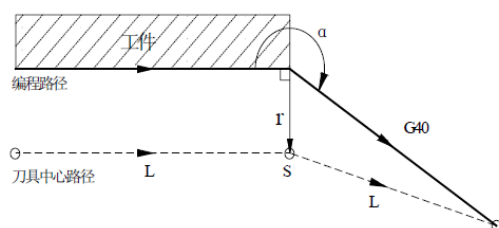
在补偿模式，当程序段满足以下任何一项条件执行时，CNC 进入补偿取消模式，这个程序段的动作称为补偿取消。

- 1、在程序中使用了 G40 代码；
- 2、执行了 M30 代码。

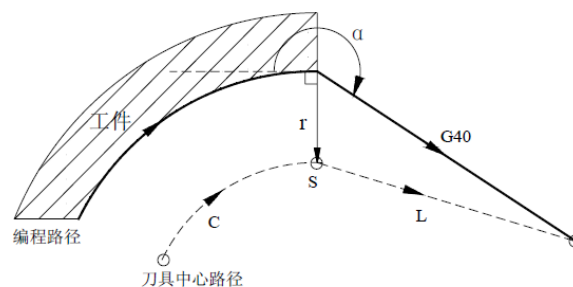
在C 刀补取消时，不可用圆弧代码 (G02 及 G03)。如果指令圆弧会产生报警且运行停止。

拐角的内侧移动 ($\alpha \geq 180^\circ$) :

1) 直线→直线

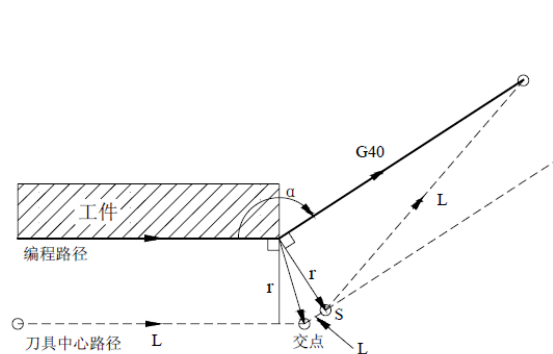


2) 圆弧→直线

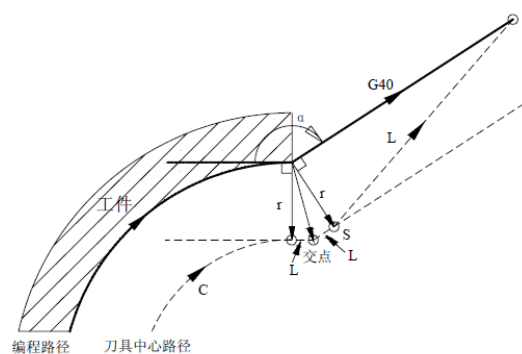


沿着拐角为钝角的外侧移动 ($180^\circ > \alpha \geq 90^\circ$) :

1) 直线→直线

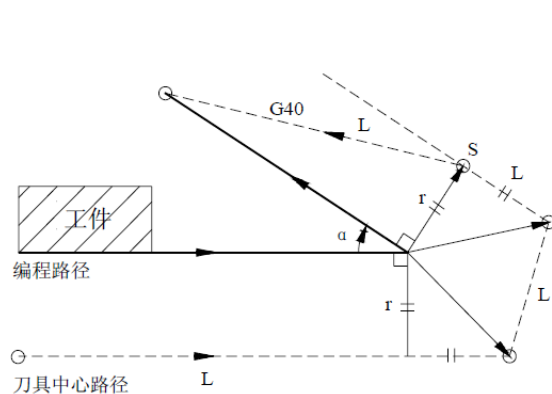


2) 圆弧→直线

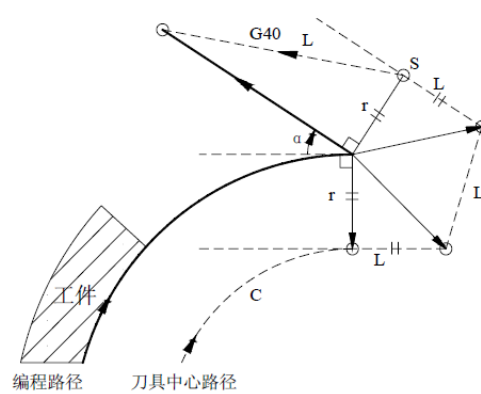


沿着拐角为锐角的外侧移动 ($\alpha < 90^\circ$) :

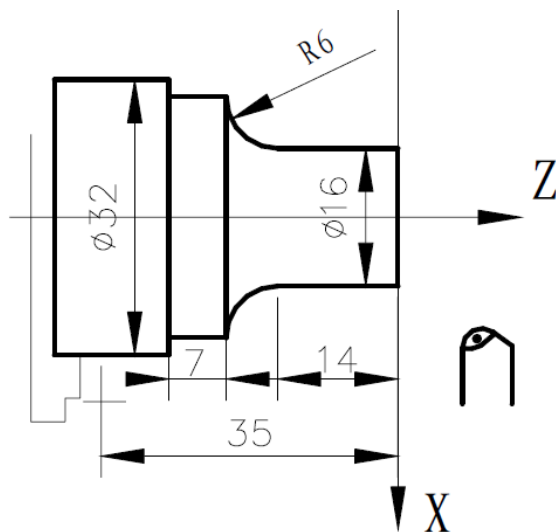
1) 直线→直线



2) 圆弧→直线



示例：在前刀座坐标系中加工下图所示零件。使用刀具号为T0101，刀尖半径 $R=2$ ，假设刀尖号 $T=3$ 。



G00 X100 Z50 M3 T0101 S600;	(定位，开主轴、换刀与执行刀补)
G42 G00 X0 Z3;	(建立刀尖半径补偿)
G01 Z0 F300;	(切削开始)
X16;	
Z-14 F200;	
G02 X28 W-6 R6;	
G01 W-7;	
X32;	
Z-35;	
G40 G00 X90 Z40;	(取消刀尖半径补偿)
G00 X100 Z50 T0100;	
M30;	

1.3.7 G50.2, 51.2(取消多边形切削, 启动多边形切削)

指令格式: G51.2 P__Q__R__ 启动多边形切削
G50.2 取消多边形切削

代码功能: G51.2指令为同步旋转工件轴和刀具轴, 以一定比例转速和相位差, 执行多边形切削加工。

同步主轴转速: 等于基础主轴转速乘以Q除以P。

同步相位差: 同步主轴相对于基础主轴以顺时针方向旋转的角度差, 如果少于R就不会做相位同步。

P: 基础主轴(工件轴)转速比率或是刀刃数, 使用内定值P=1(整数, 范围1~999)。

Q: 同步主轴(刀具轴)转速比率或是边形数, 使用内定值Q=1(整数, 范围1~999)。

R: 同步相位差(范围: $0^{\circ} \sim 359.999^{\circ}$)

注: P跟Q只能下达整数值, 若有非整数的需求, 需自行整数化。例如: 若P和Q的使用比为1: 2.5, 程序里需下达G51.2 P2 Q5。

示例:

M03 S1000;	(主轴正转, 工件轴转速1000RPM)
M24 S500;	(刀具轴反转, 刀具轴转速500RPM)
G51.2 P1 Q2 R60;	(刀具轴会同步到转速2000RPM)
G01 X50;	(进刀)
G04 X5;	
G01 X0;	(退刀)
G50.2;	(取消多边形切削)
G51.2 P1 Q3 R180;	(刀具轴会同步转速到3000RPM)
G01 X50;	(进刀)
G4 X5;	
G01 X0;	(退刀)
G50.2;	(取消多边形循环)
M5;	(主轴停止)
M25;	(刀具轴停止)
M30;	(程序结束)

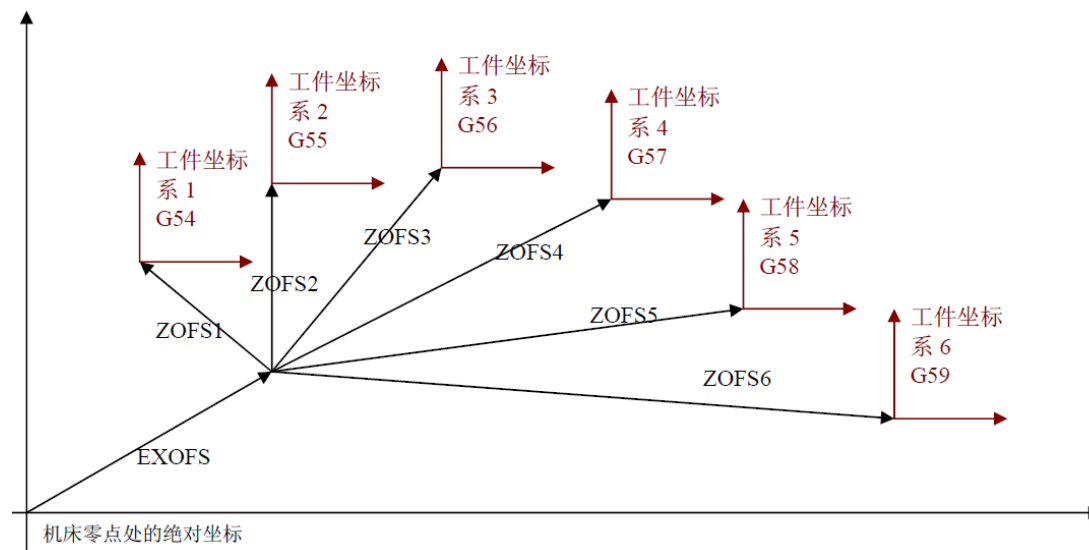
1.3.8 G54~G59(工件坐标系设定)

指令格式： G54 工件坐标系 1
 G55 工件坐标系 2
 G56 工件坐标系 3
 G57 工件坐标系 4
 G58 工件坐标系 5
 G59 工件坐标系 6

代码功能：在程序中指定G54~G59 中的一个G代码，可以从工件坐标系1~6 中选择一个。指定了工件坐标系之后，程序段中指定的点就处在指定的工件坐标系中，直到设定新的工件坐标。

代码说明：G54~G59 代码是模态代码

在系统中，每个坐标系都对应一个工件坐标系零点偏移值的参数ZOFS1~ZOFS6，与原工件坐标的关系如下：



EXOFS：外部工件坐标零点（加在每个工件坐标零点偏移值上，值在刀补->工件坐标系中设置）

ZOFSn：ZOFS1~ZOFS6 各工件坐标系零点偏移值（值在刀补->工件坐标系中设置），机械回零后设置绝对坐标系，在绝对坐标系设置完成后，再偏移EXOFS与ZOFSn 设置值。每次回零完成后的当前工件坐标的设置如下：新工件坐标 = 当前绝对坐标 - (ZOFSn + EXOFS)

G54~G59 工件坐标系的切换

通过指定G54~G59 来切换6 个工件坐标系，使系统可以工作在不同坐标系下。不同坐标系切换时，当前位置的绝对坐标的变化量为新旧工件坐标系原点的偏移量。即：新绝对坐标 = 当前绝对坐标 - (ZOFS新 - ZOFS 旧)

示例：

回零后设置绝对坐标系且0号刀偏平移工件坐标系无效时。

示例：

回零后的绝对坐标为 (0,0)；EXOFS=(0,0)；ZOFS1=(-10,-10)；

则在 G54 下回零完成后的绝对坐标值：0 - (-10 + 0) = 10；

1.3.9 G65(单一宏程序呼叫)

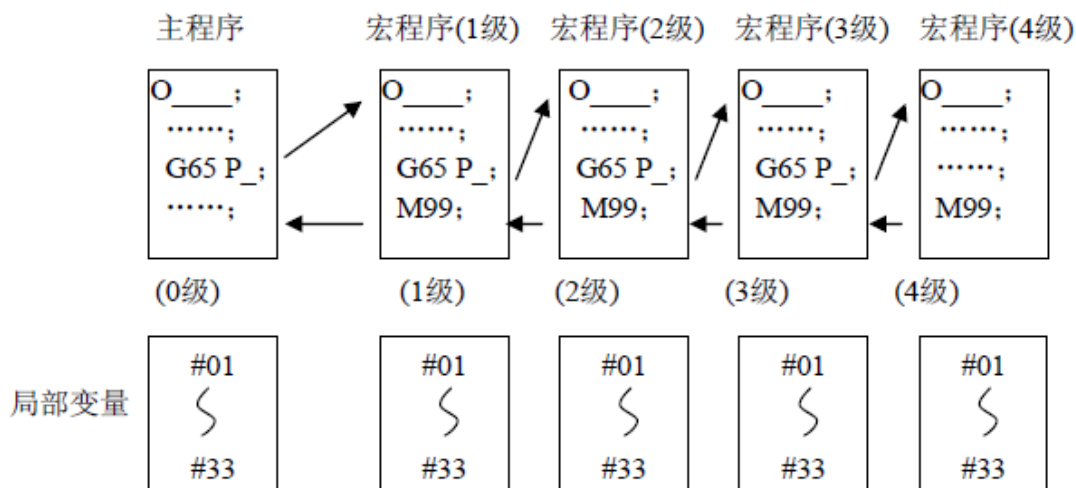
指令格式: G65 P_ L_ <自变量>_;

代码功能: 以地址P指定的宏程序被调用, 自变量(数据)传递到用户宏程序体中

代码说明: P : 被调用的宏程序号

L : 被调用的次数(省略则默认为1, 可以指定从1到9999的重复次数)

<自变量> — 被传送到宏程序中的数据,其值被赋给相应的局部变量。



示例:

G65 P9010 A1 B2 C3 I14 J15 I6 J7 K9 K11 K12 J30;

将调用程序 O9010, 同时将自变量 A1 等传递到局部变量中。对应关系如下:

#1=1, #2=2, #3=3, #4=14, #5=15, #7=6, #8=7, #6=9, #9=11, #12=12, #11=30;

1.4.0 G66（模式宏程序呼叫）/ G67（模式宏程序呼叫取消）

指令格式：G66 P_L_ <自变量>_;

代码说明：P：被调用的宏程序号；

L：被调用的次数（省略则默认为1，可以指定从1 到9999 的重复次数）；

<自变量> —— 被传送到宏程序中的数据,其值被赋给相应的局部变量。

嵌套调用：G66 调用可以有四级嵌套。

模态调用的相关说明：

1、执行 G66 程序段的作用是声明宏程序模态调用，本段并不调用宏程序。

2、执行完有 G00、G01、G02、G03代码的程序段后（声明的宏程序模态调用后取消模态调用之前）将再次调用宏程序。

3、每次调用都会按自变量更新局部变量的值。

4、G65 调用会自动取消 G66 模态

G67（模式宏程序呼叫取消）

代码格式：G67;

代码说明：取消G66 模态宏程序调用；

示例：

O2005 () ;

G00 X100 Z50;

G66 P0100 L2 A2 B20 C20 I30 J20 K20; 执行此程序段时将声明宏程序模态调用（并不调用宏程序）

G01 X80 Z50; 在执行完本段后，将执行先前声明的宏程序模态调用（并且按自变量更新局部变量）

G67; G66模态调用取消

G01 X20 Z50; 在执行完本段后，不再调用P0100号程序

M30;

1.4.1 G20、G21（英制单位加工 公制单位加工）

指令格式： G20; (英制输入)

G21; (公制输入)

代码说明: CNC数控系统的输入单位有两种: 公制单位, 毫米(mm), 英制单位, 英寸(inch)。

注意事项:

公英制切换后, 工件坐标原点偏移量、刀具数据、系统参数、与参考点位置依然正确。系统会自动处理单位转换问题。在公英制转换后, 下面操作单位会随着变动:

显示坐标、速率单位;

增量寸动单位

MPG寸动单位

1.4.2 G70 精加工循环

指令格式: G70 P(ns) Q(nf)

代码功能: 刀具从起点位置沿着ns~nf程序段给出的工件精加工轨迹进行精加工。在G71、G72或G73进行粗加工后,用G70代码进行精车,单次完成精加工余量的切削。G70循环结束时,刀具返回到起点并执行G70程序段后的下一个程序段。

ns: 精车轨迹的第一个程序段的程序段号;

nf: 精车轨迹的最后一个程序段的程序段号;

G70 代码轨迹由ns~nf 之间程序段的编程轨迹决定。ns、nf 在G70~G73 程序段中的相对位置关系如下:

```

. . . . .
G71/G72/G73 .....;
N__(ns) . . . . .
. . . . .
    · F
    · S
    ·
    ·
N__(nf).....
    ·
    ·
G70 P(ns) Q(nf);
. . .

```

} 精加工路线程序段群

注意事项:

1. G70必须在ns~nf 程序段后编写。
2. 执行G70精加工循环时, ns~nf 程序段中的F、S、T代码有效。
3. G96、G97、G98、G99、G40、G41、G42代码在执行G70精加工循环时有效。
4. 在G70代码执行过程中,可以停止自动运行并手动移动,但要再次执行G70循环时,必须返回到手动移动前的位置。如果不返回就继续执行,后面的运行轨迹将错位。
5. 在录入方式中不能执行G70代码,否则产生报警。
6. 在同一程序中需要多次使用复合循环代码时, ns~nf 不允许有相同程序段号。
7. 退刀点要尽量高或低,避免退刀碰到工件。

1.4.3 G71 横向（外径）粗车削循环

指令格式：G71 U(Δd) R(e) F_S_T_；

G71 P(ns) Q(nf) U(Δu) W(Δw)；

代码说明：G71代码分为三个部分：

(1) 给定粗车时的切削量、退刀量和切削速度、主轴转速、刀具功能的程序段；

(2) 给定定义精车轨迹的程序段区间、精车余量的程序段；

(3) 定义精车轨迹的若干连续的程序段，执行G71时，这些程序段仅用于计算粗车的轨迹，实际并未被执行。系统根据精车轨迹、精车余量、进刀量、退刀量等数据自动计算粗加工路线，沿与Z轴平行的方向切削，通过多次进刀→切削→退刀的切削循环完成工件的粗加工。G71的起点和终点相同。本代码适用于非成型毛坯(棒料)的成型粗车。

相关定义：

精车轨迹：由代码 (ns~nf程序段)给出的工件精加工轨迹，精加工轨迹的起点(即ns程序段的起点)与G71的起点、终点相同，简称A点；精加工轨迹的第一段(ns程序段)只能是X轴的快速移动或切削进给，ns程序段的终点简称B点；精加工轨迹的终点(nf程序段的终点)简称C点。精车轨迹为A点→B点→C点。

粗车轮廓：精车轨迹按精车余量(Δu 、 Δw)偏移后的轨迹，是执行G71形成的轨迹轮廓。精加工轨迹的A、B、C点经过偏移后对应粗车轮廓的A'、B'、C'点，G71代码最终的连续切削轨迹为B'点→C'点。

Δd ：粗车时X轴的切削量，取值范围0.001~99.999 (单位：mm/inch，半径值)，无符号，进刀方向由ns程序段的移动方向决定。U(Δd)执行后，指定值 Δd 保持。未输入U(Δd)时，以数据参数037号的值作为进刀量。

e：粗车时X轴的退刀量，取值范围0~99.999(单位：mm/inch，半径值)，无符号，退刀方向与进刀方向相反，R(e)执行后，指定值e保持。未输入R(e)时，以数据参数038号的值作为退刀量。

ns：精车轨迹的第一个程序段的程序段号；

nf：精车轨迹的最后一个程序段的程序段号。

Δu ：X轴的精加工余量，取值范围-99999.999~99999.999 (直径,单位：mm/inch，有符号)，粗车轮廓相对于精车轨迹的X轴坐标偏移，即：A'点与A点X轴绝对坐标的差值。U(Δu)未输入时，系统按 $\Delta u=0$ 处理，即：粗车循环X轴不留精加工余量。

Δw ：Z轴的精加工余量，取值范围-99999.999~99999.999 (单位：mm/inch,有符号)，粗车轮廓相对于精车轨迹的Z轴坐标偏移，即：A'点与A点Z轴绝对坐标的差值。W(Δw)未输入时，系统按 $\Delta w=0$ 处理，即：粗车循环Z轴不留精加工余量。

F：切削进给速度；S：主轴转速；T：刀具号、刀具偏置号。

M、S、T、F：可在第一个G71 代码或第二个G71 代码中，也可在ns~nf 程序中指定。在G71 循环中，ns~nf 间程序段号的M、S、T、F 功能都无效，仅在G70 精车循环的程序段中才有效。

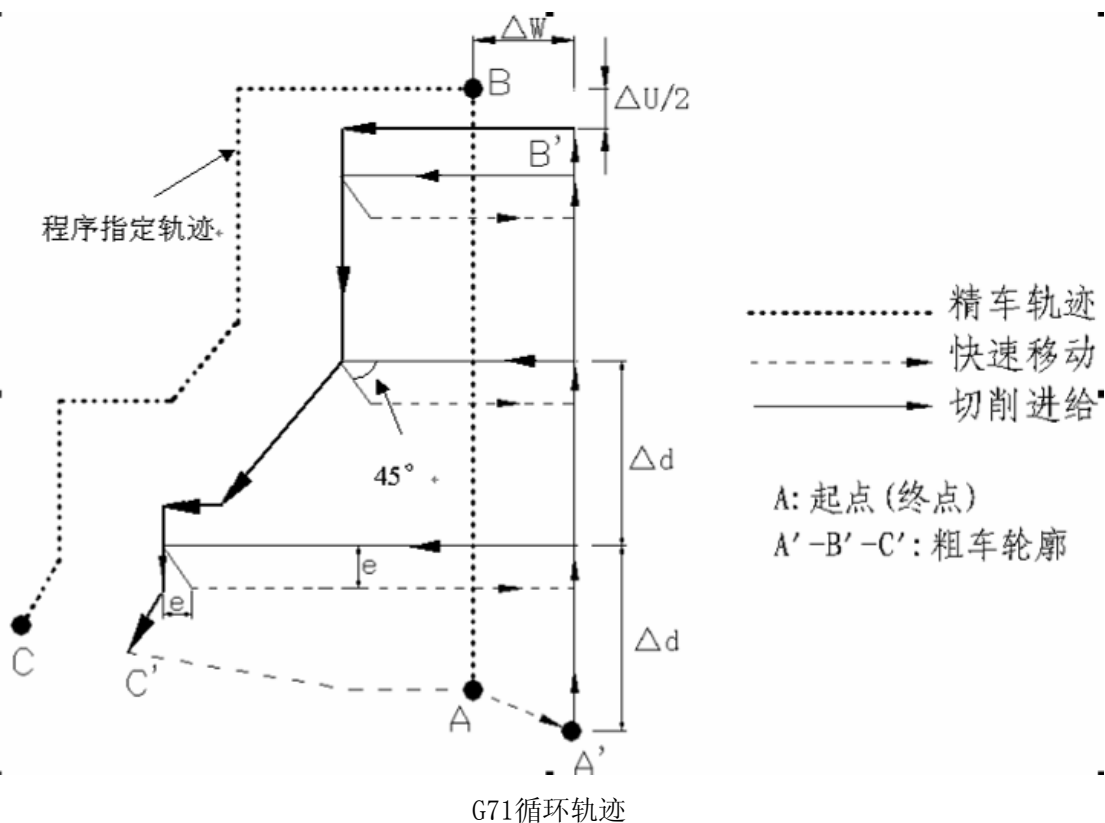
代码执行过程：

- ① 从起点A 点快速移动到A'点，X 轴移动 Δu 、Z 轴移动 Δw ；
- ② 从A'点X 轴移动 Δd (进刀)，ns 程序段是G0 时按快速移动速度进刀，ns 程序段是G1 时按G71 的切削进给速度F 进刀，进刀方向与A 点→B 点的方向一致；
- ③ Z 轴切削进给到粗车轮廓，进给方向与B 点→C 点Z 轴坐标变化一致；
- ④ X 轴、Z 轴按切削进给速度退刀e(45°直线)，退刀方向与各轴进刀方向相反；
- ⑤ Z 轴以快速移动速度退回到与A'点Z 轴绝对坐标相同的位置；

⑥ 如果X轴再次进刀($\Delta d+e$)后, 移动的终点仍在A'点→B'点的联机中间(未达到或超出B'点), X轴再次进刀($\Delta d+e$), 然后执行③; 如果X轴再次进刀($\Delta d+e$)后, 移动的终点到达B'点或超出了A'点→B'点的联机, X轴进刀至B'点, 然后执行⑦;

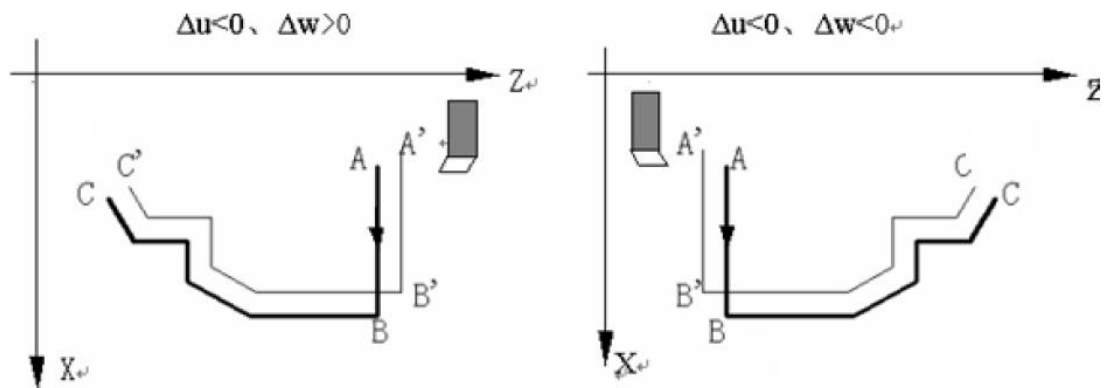
⑦ 沿粗车轮廓从B'点切削进给至C'点;

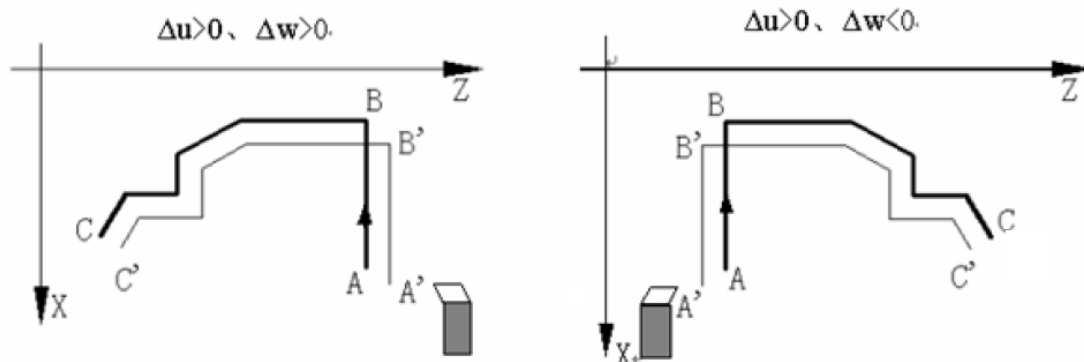
⑧ 从C'点快速移动到A点, G71循环执行结束, 程序跳转到nf程序段的下一个程序段执行。



留精车余量时坐标偏移方向:

Δu 、 Δw 反应了精车时坐标偏移和切入方向, 按 Δu 、 Δw 的符号有四种不同组合, 图中B→C为精车轨迹, B'→C'为粗车轮廓, A为起刀点。

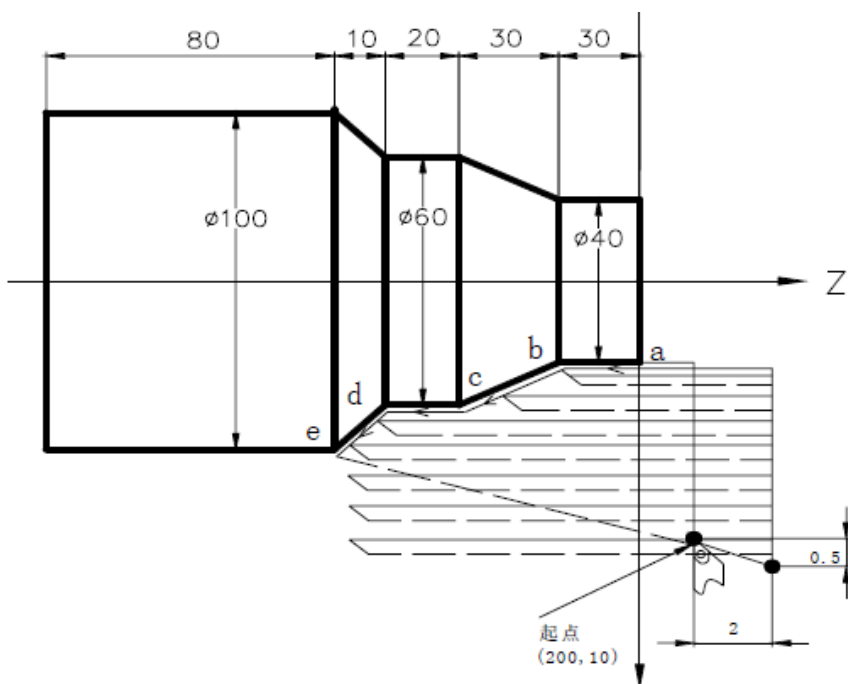




注意事项:

1. ns 程序段只能是G00、G01代码。
2. 精车轨迹(ns~nf 程序段), Z轴尺寸必须是单调变化(一直增大或一直减小), X轴尺寸也必须是单调变化。
3. ns~nf 程序段必须紧跟在G71程序段后编写。如果在G71程序段前编写, 系统自动搜索到ns~nf程序段并执行, 执行完成后, 按顺序执行G71程序段的下一程序, 因此会引起重复执行G71程序段。
4. 执行G71时, ns~nf 程序段仅用于计算粗车轮廓, 程序段并未被执行。ns~nf 程序段中的F、S、T代码在执行G71循环时无效; 执行G70精加工循环时, ns~nf程序段中的F、S、T代码有效。
5. ns~nf程序段中, 只能有G功能: G00、G01、G02、G03、G04、G96、G97、G98、G99、G40、G41、G42代码; 不能有子程序调用代码(如M98/M99)。
6. G96、G97、G98、G99、G40、G41、G42、G04代码在执行G71循环中无效, 执行G70精加工循环时有效。
7. 在G71代码执行过程中, 可以停止自动运行并手动移动, 但要再次执行G71循环时, 必须返回到手动移动前的位置。如果不返回就继续执行, 后面的运行轨迹将错位。
8. Δd , Δu 都用同一地址U指定, 其区分是根据该程序段有无指定P, Q代码。
9. 在录入方式中不能执行G71代码, 否则产生报警。
10. 在同一程序中需要多次使用复合循环代码时, ns~nf不允许有相同程序段号。
11. 退刀点要尽量高或低, 避免退刀碰到工件。

示例:



程序: O0004;

G00 X200 Z10 M3 S800;	(逆时针转, 转速800r/min)
G71 U2 R1 F200;	(每次切深4mm, 退刀2mm,[直径])
G71 P80 Q120 U1 W2;	(对a---e粗车加工, 余量X方向1mm, Z方向2mm)
N80 G00 X40 S1200;	(定位)
G01 Z-30 F100 ;	(a→b)
X60 W-30;	(b→c) 精加工路线a→b→c→d→e程序段
W-20;	(c→d)
N120 X100 W-10;	(d→e)
G70 P80 Q120;	(对a---e精车加工)
M30;	(程序结束)

1.4.4 G72径向（端面）粗车削循环

指令格式： G72 W(Δd) R(e) F_ S_ T_ ;
G72 P(ns) Q(nf) U(Δu) W(Δw);

代码说明：G72代码分为三个部分：

- (1) 给定粗车时的切削量、退刀量和切削速度、主轴转速、刀具功能的程序段；
- (2) 给定定义精车轨迹的程序段区间、精车余量的程序段；
- (3) 定义精车轨迹的若干连续的程序段，执行G72时，这些程序段仅用于计算粗车的轨迹，实际并未被执行。系统根据精车轨迹、精车余量、进刀量、退刀量等数据自动计算粗加工路线，沿与X轴平行的方向切削，通过多次进刀→切削→退刀的切削循环完成工件的粗加工，G72的起点和终点相同。本代码适用于非成型毛坯(棒料)的成型粗车。

相关定义：

精车轨迹：由代码 (ns~nf程序段)给出的工件精加工轨迹，精加工轨迹的起点(即ns程序段的起点)与G72的起点、终点相同，简称A点；精加工轨迹的第一段(ns程序段)只能是Z轴的快速移动或切削进给，ns程序段的终点简称B点；精加工轨迹的终点(nf程序段的终点)简称C点。精车轨迹为A点→B点→C点。

粗车轮廓：精车轨迹按精车余量(Δu 、 Δw)偏移后的轨迹，是执行G72形成的轨迹轮廓。精加工轨迹的A、B、C点经过偏移后对应粗车轮廓的A'、B'、C'点，G72代码最终的连续切削轨迹为B'点→C'点。

Δd ：粗车时Z轴的切削量，取值范围0.001~99.999(单位：mm/inch)，无符号，进刀方向由ns程序段的移动方向决定。W(Δd)执行后，指定值 Δd 保持，未输入W(Δd)时，以数据参数037号的值作为进刀量。

e：粗车时Z轴的退刀量，取值范围0~99.999(单位：mm/inch)，无符号，退刀方向与进刀方向相反，R(e)执行后，指定值e保持，未输入R(e)时，以数据参数038号的值作为退刀量。

ns：精车轨迹的第一个程序段的程序段号。

nf：精车轨迹的最后一个程序段的程序段号。

Δu ：粗车时X轴留出的精加工余量，取值范围-99999.999~99999.999(粗车轮廓相对于精车轨迹的X轴坐标偏移，即：A'点与A点X轴绝对坐标的差值，单位：mm/inch，直径，有符号)。

Δw ：粗车时Z轴留出的精加工余量，取值范围-99999.999~99999.999(粗车轮廓相对于精车轨迹的Z轴坐标偏移，即：A'点与A点Z轴绝对坐标的差值，单位：mm/inch，有符号)。

F：切削进给速度；**S：**主轴转速；**T：**刀具号、刀具偏置号。

M、S、T、F：可在第一个G72代码或第二个G72代码中，也可在ns~nf程序中指定。在G72循环中，ns~nf间程序段号的M、S、T、F功能都无效，仅在G70精车循环的程序段中才有效。

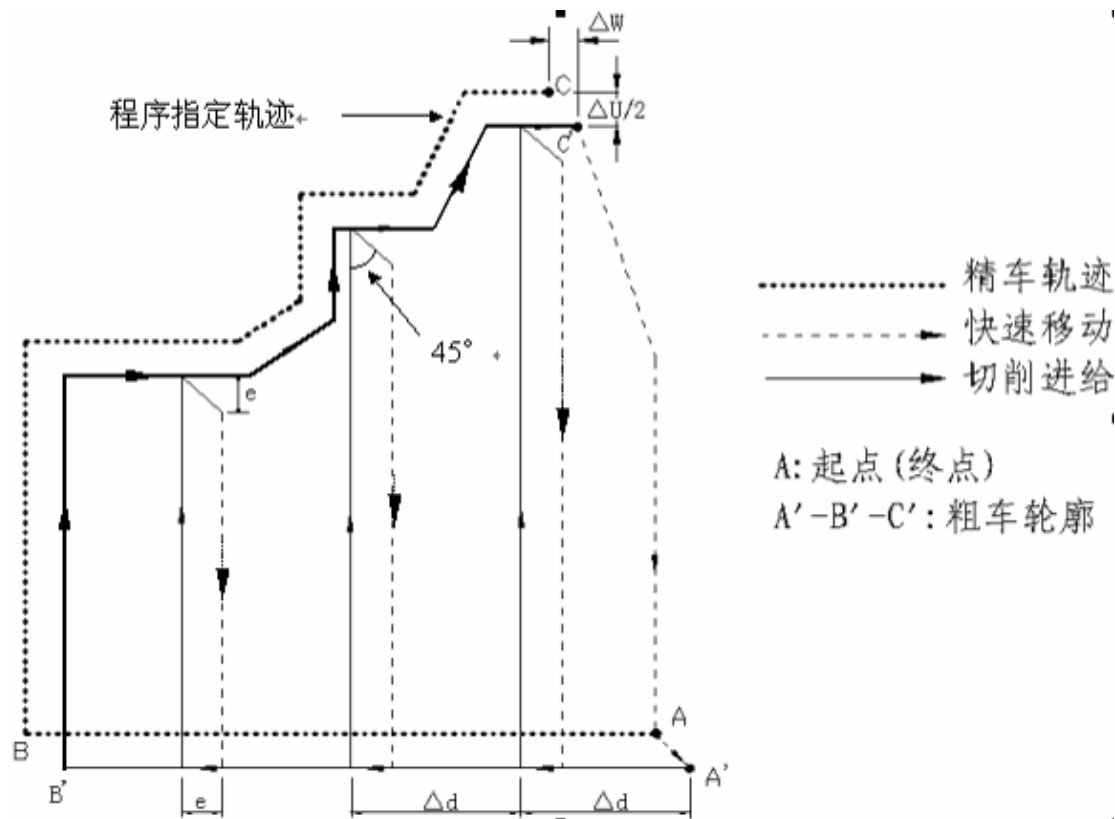
代码执行过程：

- ① 从起点A点快速移动到A'点，X轴移动 Δu 、Z轴移动 Δw ；
- ② 从A'点Z轴移动 Δd (进刀)，ns程序段是G0时按快速移动速度进刀，ns程序段是G1时按G72的切削进给速度F进刀，进刀方向与A点→B点的方向一致；
- ③ X轴切削进给到粗车轮廓，进给方向与B点→C点X轴坐标变化一致；
- ④ X轴、Z轴按切削进给速度退刀e(45°直线)，退刀方向与各轴进刀方向相反；
- ⑤ X轴以快速移动速度退回到与A'点Z轴绝对坐标相同的位置；
- ⑥ 如果Z轴再次进刀($\Delta d+e$)后，移动的终点仍在A'点→B'点的联机中间(未达到或超

出B'点), Z 轴再次进刀($\Delta d+e$), 然后执行③; 如果Z 轴再次进刀($\Delta d+e$)后, 移动的终点到达B'点或超出了A'点→B'点的联机, Z 轴进刀至B'点, 然后执行⑦;

⑥ 沿粗车轮廓从B'点切削进给至C'点;

⑧从C'点快速移动到A 点, G72 循环执行结束, 程序跳转到nf 程序段的下一个程序段执行。



注意事项:

1. ns~nf 程序段必须紧跟在G72程序后编写。如果在G72程序段前编写, 系统自动搜索到ns~nf程序段并执行, 执行完成后, 按顺序执行nf 程序段的下一程序, 因此会引起重复执行ns~nf 程序段。

2. 执行G72时, ns~nf 程序段仅用于计算粗车轮廓, 程序段并未被执行。ns~nf 程序段中的F、S、T代码在执行G72循环时无效。执行G70精加工循环时, ns~nf程序段中的F、S、T代码有效。

3. ns 程序段只能是不含X(U)代码字的G00、G01代码, 否则报警。

4. 精车轨迹(ns~nf程序段), X轴、Z轴的尺寸都必须是单调变化(一直增大或一直减小)。

5. ns~nf程序段中, 只能有G功能: G00、G01、G02、G03、G04、G96、G97、G98、G99、G40、G41、G42代码; 不能有子程序调用代码(如M98/M99)。

6. G96、G97、G98、G99、G40、G41、G42代码在执行G72循环中无效, 执行G70精加工循环时有效。

7. 在G72代码执行过程中, 可以停止自动运行并手动移动, 但要再次执行G72循环时, 必须返回到手动移动前的位置。如果不返回就继续执行, 后面的运行轨迹将错位。

8. Δd , Δw 都用同一地址W指定, 其区分是根据该程序段有无指定P, Q代码字。

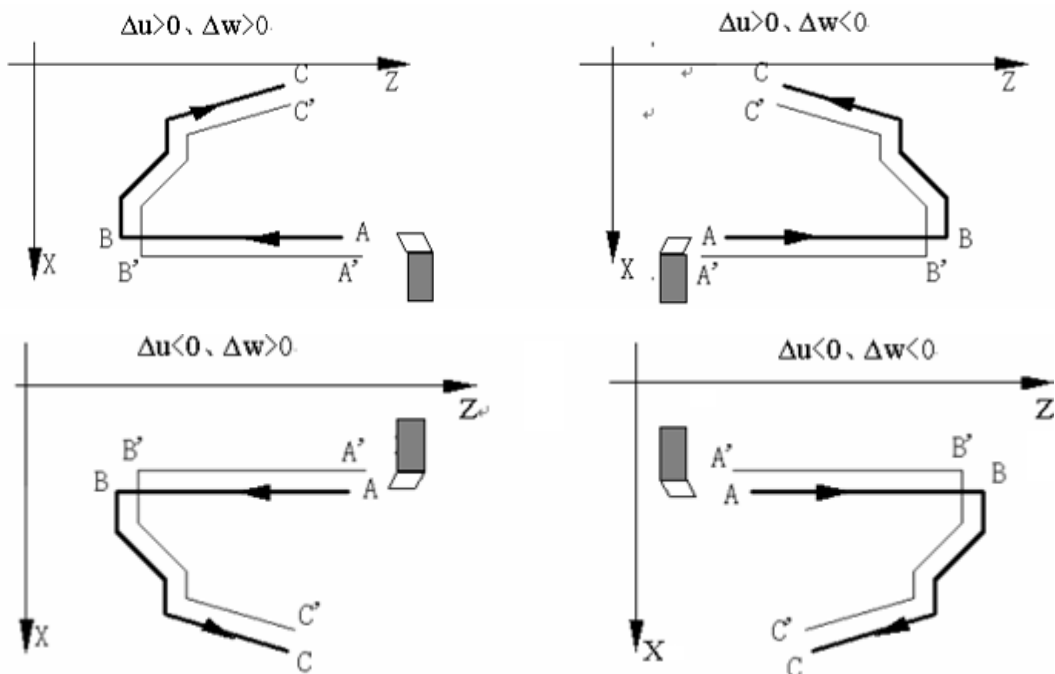
9. 在同一程序中需要多次使用复合循环代码时, ns~nf不允许有相同程序段号。

10. 在录入方式中不能执行G72代码, 否则产生报警。

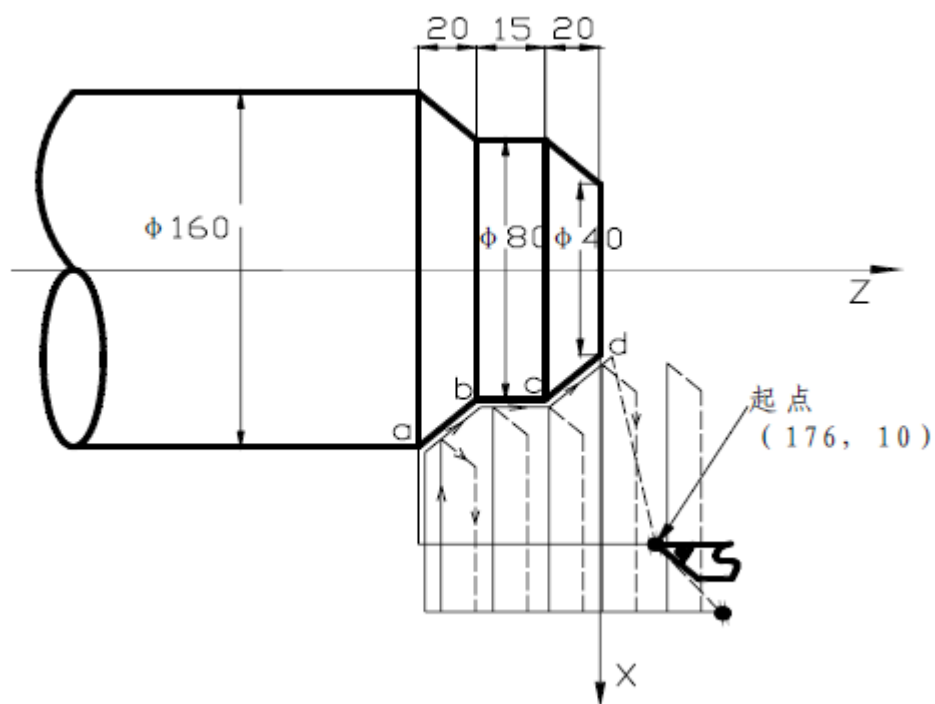
11. 退刀点要尽量高或低，避免退刀碰到工件。

留精车余量时坐标偏移方向：

Δu 、 Δw 反应了精车时坐标偏移和切入方向，按 Δu 、 Δw 的符号有四种不同组合，见图3-29，图中： $B \rightarrow C$ 为精车轨迹， $B' \rightarrow C'$ 为粗车轮廓，A为起刀点。



示例：



程序: O0005;

G00 X176 Z10 M03 S500 (换2号刀, 执行2号刀偏, 逆时针转, 转速500)

G72 W2.0 R0.5 F300; (进刀量2mm, 退刀量0.5mm)

G72 P10 Q20 U0.2 W0.1; (对a--d粗车, X留0.2mm, Z留0.1mm余量)

N10 G00 Z-55 S800 ; (快速移动)

G01 X160 F120; (进刀至a点)

X80 W20; (加工a—b)

W15; (加工b—c)

N20 X40 W20 ; (加工c—d)

G70 P010 Q020 M30; (精加工a—d)

1.4.5 G73 (成形轮廓粗车削循环)

指令格式: G73 U(Δi) W(Δk) R(d) F_ S_ T_ ;

G73 P(ns) Q(nf) U(Δu) W(Δw);

代码说明: : G73代码分为三个部分:

- (1) 给定退刀量、切削次数和切削速度、主轴转速、刀具功能的程序段;
- (2) 给定定义精车轨迹的程序段区间、精车余量的程序段;
- (3) 定义精车轨迹的若干连续的程序段, 执行G73时, 这些程序段仅用于计算粗车的轨迹, 实际并未被执行。系统根据精车余量、退刀量、切削次数等数据自动计算粗车偏移量、粗车的单次进刀量和粗车轨迹, 每次切削的轨迹都是精车轨迹的偏移, 切削轨迹逐步靠近精车轨迹, 最后一次切削轨迹为按精车余量偏移的精车轨迹。G73 的起点和终点相同, 本代码适用于成型毛坯的粗车。

相关定义:

精车轨迹: 由代码 (ns~nf程序段)给出的工件精加工轨迹, 精加工轨迹的起点(即ns程序段的起点)与G73的起点、终点相同, 简称A点; 精加工轨迹的第一段(ns程序段)的终点简称B点; 精加工轨迹的终点(nf程序段的终点)简称C点。精车轨迹为A点→B点→C点。

粗车轨迹: 为精车轨迹的一组偏移轨迹, 粗车轨迹数量与切削次数相同。坐标偏移后精车轨迹的A、B、C点分别对应粗车轨迹的A_n、B_n、C_n点(n为切削的次数, 第一次切削表示为A₁、B₁、C₁点, 最后一次表示为A_d、B_d、C_d点)。第一次切削相对于精车轨迹的坐标偏移量为 ($\Delta i \times 2 + \Delta u$, $\Delta w + \Delta k$)(按直径编程表示), 最后一次切削相对于精车轨迹的坐标偏移量为(Δu , Δw), 每一次切削相对于上一次切削轨迹的坐标偏移量为:

$$\left(-\frac{\Delta i \times 2}{d-1}, -\frac{\Delta k}{d-1} \right)$$

Δi : X轴粗车退刀量, 取值范围 $\pm 99999999 \times$ 最小输入增量(单位: mm/inch, 半径值, 有符号), Δi 等于A₁点相对于A_d点的X轴坐标偏移量(半径值), 粗车时X轴的总切削量(半径值)等于 $|\Delta i|$, X轴的切削方向与 Δi 的符号相反: $\Delta i > 0$, 粗车时向X轴的负方向切削。 Δi 指定值执行后保持, 未输入U(Δi)时, 以数据参数039号的值作为X轴粗车退刀量。

Δk : Z轴粗车退刀量, 取值范围-99999.999~99999.999(单位: mm/inch, 有符号), Δk 等于A₁点相对于A_d点的Z轴坐标偏移量, 粗车时Z轴的总切削量等于 $|\Delta k|$, Z轴的切削方向与 Δk 的符号相反: $\Delta k > 0$, 粗车时向Z轴的负方向切削。 Δk 指定值执行后保持, 未输入W(Δk)时, 以数据参数040号的值作为Z轴粗车退刀量。

d: 切削的次数, 取值范围1~9999(单位: 次), R5表示5次切削完成封闭切削循环。R(d)指定值执行后保持, 未输入R(d)时, 以数据参数041号的值作为切削次数。如果切削次数为1, 系统将按2次切削完成封闭切削循环。

ns: 精车轨迹的第一个程序段的程序段号。

nf: 精车轨迹的最后一个程序段的程序段号。

Δu : X轴的精加工余量, 取值范围-99999.999~99999.999(单位: mm/inch, 直径, 有符号), 最后一次粗车轨迹相对于精车轨迹的X轴坐标偏移, 即: A₁点相对于A点X轴绝对坐标的差值。 $\Delta u > 0$, 最后一次粗车轨迹相对于精车轨迹向X轴的正方向偏移。未输入U(Δu)时, 系统按 $\Delta u = 0$ 处理, 即: 粗车循环X轴不留精加工余量。

Δw : Z轴的精加工余量, 取值范围-99999.999~99999.999(单位: mm/inch, 有符号), 最后一次粗车轨迹相对于精车轨迹的Z轴坐标偏移, 即: A₁点相对于A点Z轴绝对坐标的差值。 $\Delta w > 0$, 最后一次粗车轨迹相对于精车轨迹向Z轴的正方向偏移。未输入W(Δw)时, 系统按 $\Delta w = 0$ 处理, 即: 粗车循环Z轴不留精加工余量。

F: 切削进给速度; S: 主轴转速; T: 刀具号、刀具偏置号。

M、S、T、F: 代码字可在第一个G73 代码或第二个G73 代码中, 也可在 $ns \sim nf$ 程序中指定。在G73循环中, $ns \sim nf$ 间程序段号的M、S、T、F 功能都无效, 仅在有G70 精车循环的程序段中才有效。

代码执行过程:

$A \rightarrow A_1$: 快速移动;

第一次粗车, $A_1 \rightarrow B_1 \rightarrow C_1$:

$A_1 \rightarrow B_1$: ns 程序段是G0 时按快速移动速度, ns 程序段是G1 时按G73 指定的切削进给速度;

$B_1 \rightarrow C_1$: 切削进给。

$C_1 \rightarrow A_2$: 快速移动;

第二次粗车, $A_2 \rightarrow B_2 \rightarrow C_2$:

$A_2 \rightarrow B_2$: ns 程序段是G0 时按快速移动速度, ns 程序段是G1 时按G73 指定的切削进给速度;

$B_2 \rightarrow C_2$: 切削进给。

$C_2 \rightarrow A_3$: 快速移动;

.....

第 n 次粗车, $A_n \rightarrow B_n \rightarrow C_n$:

$A_n \rightarrow B_n$: ns 程序段是G0 时按快速移动速度, ns 程序段是G1 时按G73 指定的切削进给速度;

$B_n \rightarrow C_n$: 切削进给。

$C_n \rightarrow A_{n+1}$: 快速移动;

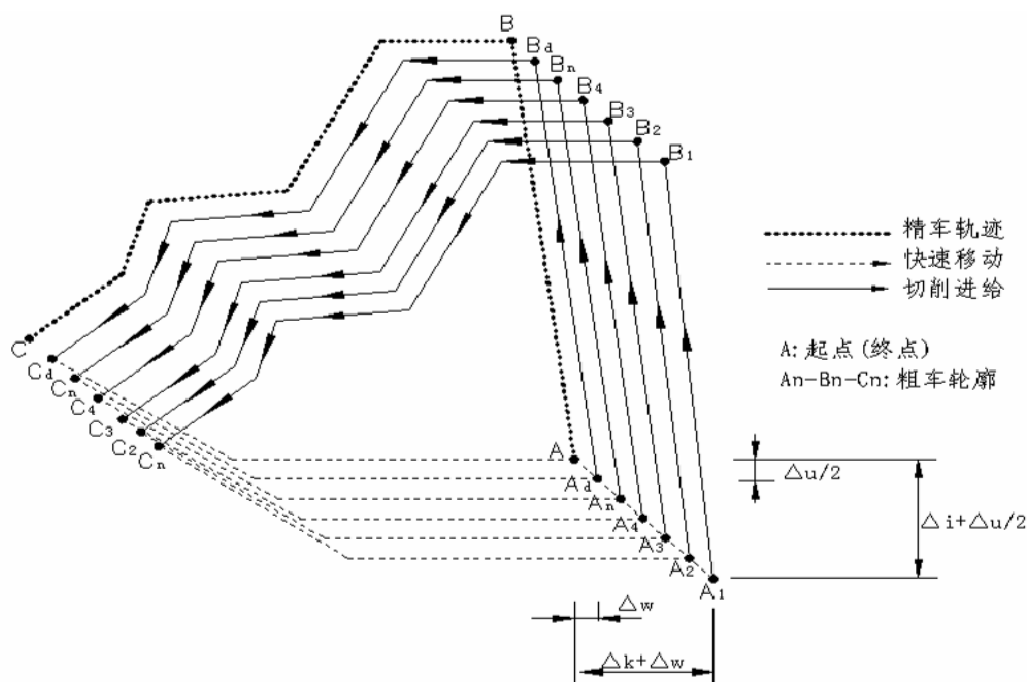
.....

最后一次粗车, $A_d \rightarrow B_d \rightarrow C_d$:

$A_d \rightarrow B_d$: ns 程序段是G0 时按快速移动速度, ns 程序段是G1 时按G73 指定的切削进给速度;

$B_d \rightarrow C_d$: 切削进给。

$C_d \rightarrow A$: 快速移动到起点;



注意事项:

1. ns~nf 程序段必须紧跟在G73程序段后编写。ns~nf 程序段如果在G73程序段前编写,系统能自动搜索到ns~nf程序段并执行,执行完成后,按顺序执行nf 程序段的下一程序,因此会引起重复执行ns~nf 程序段。

2. 执行G73时, ns~nf程序段仅用于计算粗车轮廓,程序段并未被执行。ns~nf程序段中的F、S、T代码在执行G73时无效。执行G70精加工循环时, ns~nf 程序段中的F、S、T代码有效。

3. ns 程序段只能是G00、G01代码。

4. ns~nf 程序段中,只能有下列G功能: G00、G01、G02、G03、G04、G96、G97、G98、G99、G40、G41、G42代码;不能有下列M功能: 子程序调用代码(如M98/M99)。

5. G96、G97、G98、G99、G40、G41、G42代码在执行G73循环中无效,执行G70精加工循环时有效。

6. 在G73代码执行过程中,可以停止自动运行并手动移动,但要再次执行G73循环时,必须返回到手动移动前的位置。如果不返回就继续执行,后面的运行轨迹将错位。

7. Δi , Δu 都用同一地址U指定, Δk , Δw 都用同一地址W指定,其区分是根据该程序段有无指定P, Q代码字。

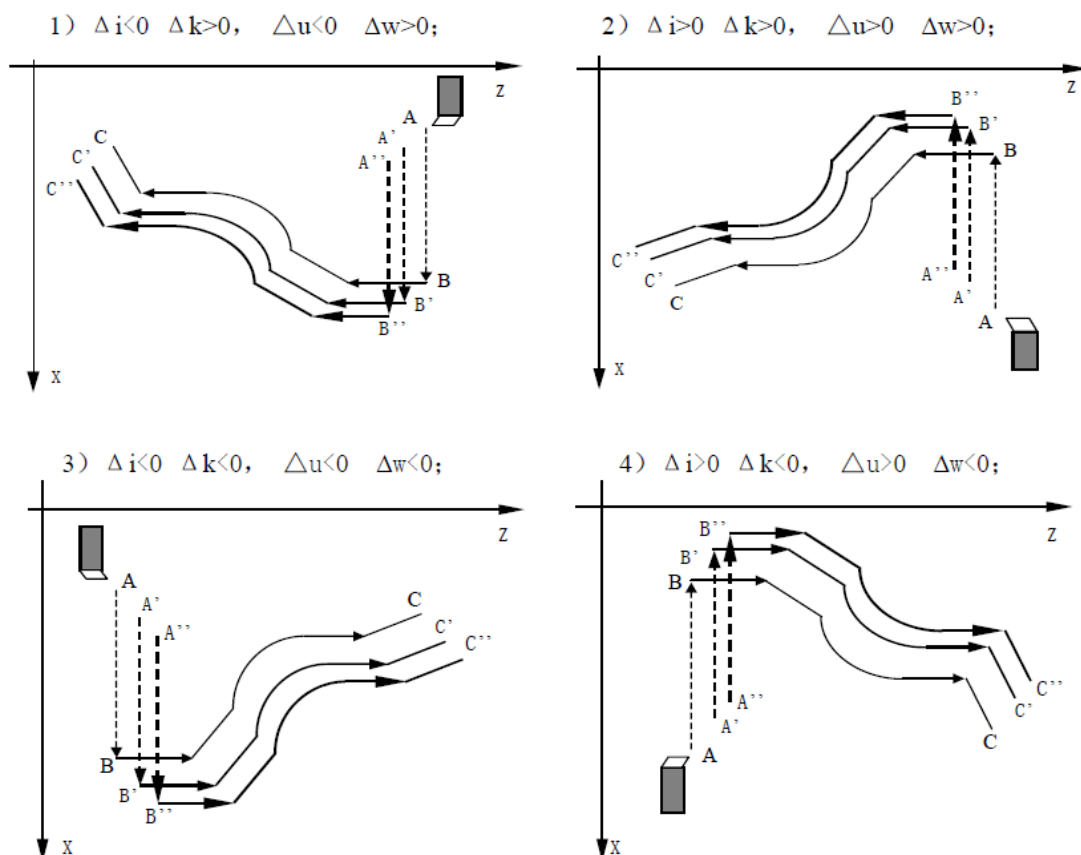
8. 在录入方式中不能执行G73代码,否则产生报警。

9. 在同一程序中需要多次使用复合循环代码时, ns~nf 不允许有相同程序段号。

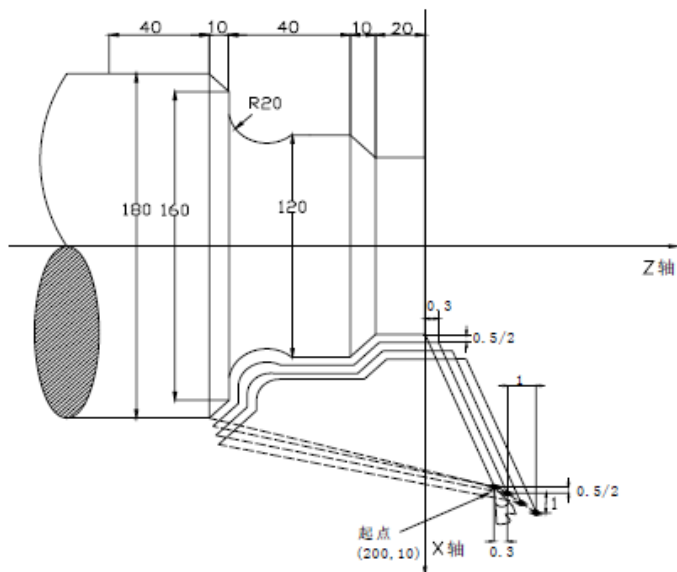
10. 退刀点要尽量高或低,避免退刀碰到工件。

留精车余量时坐标偏移方向:

Δi 、 Δk 反应了粗车时坐标偏移和切入方向, Δu 、 Δw 反应了精车时坐标偏移和切入方向; Δi 、 Δk 、 Δu 、 Δw 可以有多种组合,在一般情况下,通常 Δi 与 Δu 的符号一致, Δk 与 Δw 的符号一致,常用有四种组合,图中: A为起刀点, B→C为工件轮廓, B'→C'为粗车轮廓, B''→C''为精车轨迹。



示例:



程序:

O0006;

G99 G00 X200 Z10 M03 S500;

(指定每转进给, 定位起点, 启动主轴)

G73 U1.0 W1.0 R3 ;

(X轴退刀2mm, Z轴退刀1mm)

G73 P14 Q19 U0.5 W0.3 F0.3 ;

(粗车, X轴留0.5mm, Z轴留0.3mm精车余量)

N14 G00 X80 Z0 ;

G01 W-20 F0.15 S600 ;

X120 W-10 ;

W-20 ;

G02 X160 W-20 R20 ;

N19 G01 X180 W-10 ;

G70 P14 Q19 M30;

(精加工)

} 精加工形状程序段

1.4.6 G74 端面（Z轴）啄式加工循环

指令格式： G74 R(e);

G74 X(U)_ Z(W)_ P(Δi) Q(Δk) R(Δd) F_ ;

代码功能：径向(X 轴)进刀循环复合轴向断续切削循环：从起点轴向(Z 轴)进给、回退、再进给……直至切削到与切削终点Z 轴坐标相同的位置，然后径向退刀、轴向回退至与起点Z 轴坐标相同的位置，完成一次轴向切削循环；径向再次进刀后，进行下一次轴向切削循环；切削到切削终点后，返回起点(G74 的起点和终点相同)，轴向切槽复合循环完成。G74 的径向进刀和轴向进刀方向由切削终点X(U)、Z(W)与起点的相对位置决定，此代码用于在工件端面加工环形槽或中心深孔，轴向断续切削起到断屑、及时排屑的作用。

相关定义：

轴向切削循环起点：每次轴向切削循环开始轴向进刀的位置，表示为 $A_n(n=1,2,3,\dots)$ ， A_n 的Z轴坐标与起点A 相同， A_n 与 A_{n-1} 的X 轴坐标的差值为 Δi 。第一次轴向切削循环起点 A_1 与起点A 为同一点，最后一次轴向切削循环起点(表示为 A_f)的X 轴坐标与切削终点相同。

轴向进刀终点：每次轴向切削循环轴向进刀的终点位置，表示为 $B_n(n=1,2,3,\dots)$ ， B_n 的Z 轴坐标与切削终点相同， B_n 的X 轴坐标与 A_n 相同，最后一次轴向进刀终点(表示为 B_f)与切削终点为同一点；

径向退刀终点：每次轴向切削循环到达轴向进刀终点后，径向退刀(退刀量为 Δd)的终点位置，表示为 $C_n(n=1,2,3,\dots)$ ， C_n 的Z 轴坐标与切削终点相同， C_n 与 A_n X 轴坐标的差值为 Δd ；

轴向切削循环终点：从径向退刀终点轴向退刀的终点位置，表示为 $D_n(n=1,2,3,\dots)$ ， D_n 的Z 轴坐标与起点相同， D_n 的X 轴坐标与 C_n 相同(与 A_n X 轴坐标的差值为 Δd)；

切削终点：X(U) Z(W) 指定的位置，最后一次轴向进刀终点 B_f 。

R(e)：每次轴向(Z 轴)进刀后的轴向退刀量，取值范围0~99.999(单位：mm)，无符号。
R(e)执行后指定值保持有效，未输入R(e)时，以数据参数042号的值作为轴向退刀量。

X：切削终点 B_f 的X 轴绝对坐标值。

U：切削终点 B_f 与起点A 的X 轴绝对坐标的差值。

Z：切削终点 B_f 的Z 轴的绝对坐标值。

W：切削终点 B_f 与起点A 的Z 轴绝对坐标的差值。

P(Δi)：单次轴向切削循环的径向(X 轴)切削量，取值范围 $0 < \Delta i \leq 9999999$ （单位：最小输入增量，直径值，无符号）。

Q(Δk)：轴向(Z 轴)切削时，Z 轴断续进刀的进刀量，取值范围 $0 < \Delta k \leq 9999999$ （单位：最小输入增量，无符号）。

R(Δd)：切削至轴向切削终点后，径向(X 轴)的退刀量，取值范围 $0 \sim 99999999 \times$ 最小输入增量(单位：mm/inch，直径值，无符号)，省略R(Δd)时，系统默认轴向切削终点后，径向(X 轴)的退刀量为0。省略X(U)和P(Δi)代码字时，默认往正方向退刀。

代码执行过程：

① 从轴向切削循环起点 A_n 轴向(Z 轴)切削进给 Δk ，切削终点Z 轴坐标小于起点Z 轴坐标时，向Z 轴负向进给，反之则向Z 轴正向进给；

② 轴向(Z 轴)快速移动退刀e，退刀方向与①进给方向相反；

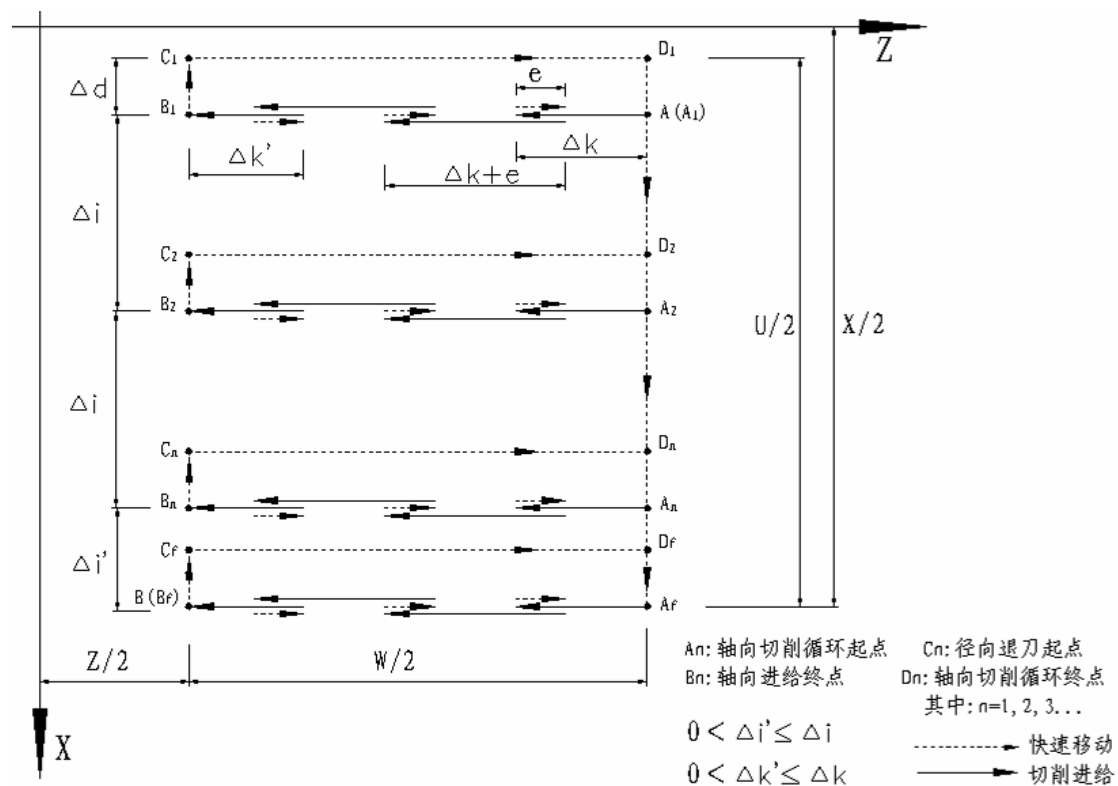
③ 如果Z 轴再次切削进给($\Delta k+e$)，进给终点仍在轴向切削循环起点 A_n 与轴向进刀终点 B_n 之间，Z 轴再次切削进给($\Delta k+e$)，然后执行②；如果Z 轴再次切削进给($\Delta k+e$)后，进给终点到达 B_n 点或不在 A_n 与 B_n 之间，Z 轴切削进给至 B_n 点，然后执行④；

④ 径向(X 轴)快速移动退刀 Δd 至 C_n 点, B_f 点(切削终点)的X 轴坐标小于A 点(起点)X 轴坐标时, 向X 轴正向退刀, 反之则向X 轴负向退刀;

⑤ 轴向(Z 轴)快速移动退刀至 D_n 点, 第 n 次轴向切削循环结束。如果当前不是最后一次轴向切削循环, 执行⑥; 如果当前是最后一次轴向切削循环, 执行⑦;

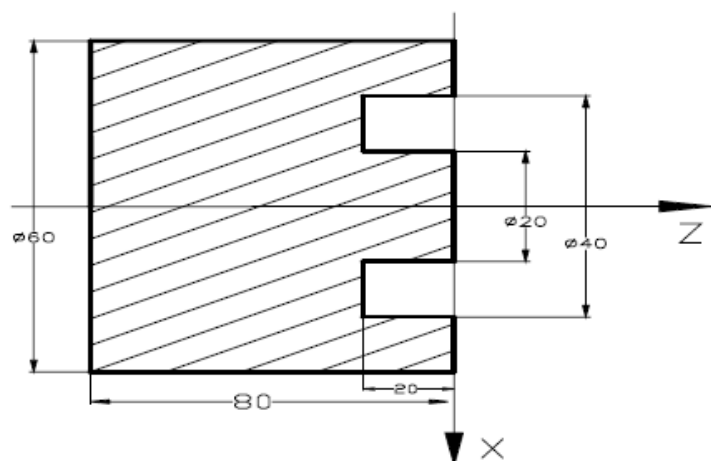
⑥ 径向(X 轴)快速移动进刀, 进刀方向与④退刀方向相反。如果X 轴进刀($\Delta d + \Delta i$)后, 进刀终点仍在A 点与 A_f 点(最后一次轴向切削循环起点)之间, X 轴快速移动进刀($\Delta d + \Delta i$), 即: $D_n \rightarrow A_{n+1}$, 然后执行①(开始下一次轴向切削循环); 如果X 轴进刀($\Delta d + \Delta i$)后, 进刀终点到达 A_f 点或不在 D_n 与 A_f 点之间, X 轴快速移动至 A_f 点, 然后执行①, 开始最后一次轴向切削循环;

⑦ X 轴快速移动返回到起点A, G74 代码执行结束。



G74运动轨迹图

示例:



程序(假设切槽刀宽度为4mm，系统的最小增量为0.001mm):

O0007;

G0 X36 Z5 M3 S500; (启动主轴，定位到加工起点，X方向加上刀具宽度)

G74 R0.5 ; (加工循环)

G74 X20 Z-20 P3000 Q5000 F50; (Z轴每次进刀5mm，退刀0.5mm，进给到终点(Z-20)后，快速返回到起点(Z5)，X轴进刀3mm，循环以上步骤继续运行)

M30; (程序结束)

1.4.7 G75 横向(X轴)啄式加工循环

指令格式: G75 R(e);

G75 X(U)_ Z(W)_ P(Δi) Q(Δk) R(Δd) F_ ;

代码功能: 轴向(Z 轴)进刀循环复合径向断续切削循环: 从起点径向(X 轴)进给、回退、再进给.....直至切削到与切削终点X 轴坐标相同的位置, 然后轴向退刀、径向回退至与起点X 轴坐标相同的位置, 完成一次径向切削循环; 轴向再次进刀后, 进行下一次径向切削循环; 切削到切削终点后, 返回起点(G75 的起点和终点相同), 径向切槽复合循环完成。G75 的轴向进刀和径向进刀方向由切削终点X(U)Z(W)与起点的相对位置决定, 此代码用于加工径向环形槽或圆柱面, 径向断续切削起到断屑、及时排屑的作用。

相关定义:

径向切削循环起点: 每次径向切削循环开始径向进刀的位置, 表示为 $A_n(n=1,2,3,\dots)$, A_n 的X轴坐标与起点A 相同, A_n 与 A_{n-1} 的Z 轴坐标的差值为 Δk 。第一次径向切削循环起点 A_1 与起点A 为同一点, 最后一次径向切削循环起点(表示为 A_f)的Z 轴坐标与切削终点相同。

径向进刀终点: 每次径向切削循环径向进刀的终点位置, 表示为 $B_n(n=1,2,3,\dots)$, B_n 的X 轴坐标与切削终点相同, B_n 的Z 轴坐标与 A_n 相同, 最后一次径向进刀终点(表示为 B_f)与切削终点为同一点;

轴向退刀终点: 每次径向切削循环到达径向进刀终点后, 轴向退刀(退刀量为 Δd)的终点位置, 表示为 $C_n(n=1,2,3,\dots)$, C_n 的X 轴坐标与切削终点相同, C_n 与 A_n Z 轴坐标的差值为 Δd ;

径向切削循环终点: 从轴向退刀终点径向退刀的终点位置, 表示为 $D_n(n=1,2,3,\dots)$, D_n 的X 轴坐标与起点相同, D_n 的Z 轴坐标与 C_n 相同(与 A_n Z 轴坐标的差值为 Δd);

切削终点: X(U) Z(W) 指定的位置, 最后一次径向进刀终点 B_f 。

R(e): 每次径向(X 轴)进刀后的径向退刀量, 取值范围0~99.999(单位: mm, 半径值), 无符号。R(e)执行后指定值保持有效, 未输入R(e)时, 以数据参数042号的值作为径向退刀量。

X: 切削终点 B_f 的X 轴绝对坐标值。

U: 切削终点 B_f 与起点A 的X 轴绝对坐标的差值。

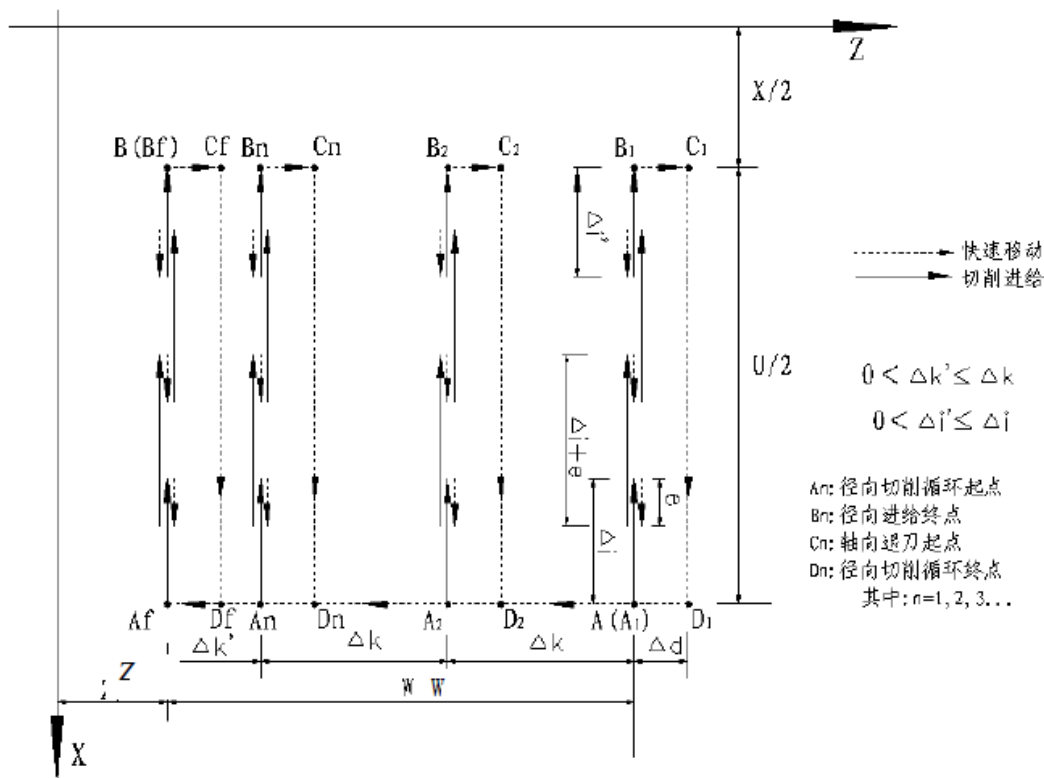
Z: 切削终点 B_f 的Z 轴的绝对坐标值。

W: 切削终点 B_f 与起点A 的Z 轴绝对坐标的差值。

P(Δi): 径向(X 轴)进刀时, X 轴断续进刀的进刀量, 取值范围 $0 < \Delta i \leq 9999999$ (单位: 最小输入增量, 无符号)。

Q(Δk): 单次径向切削循环的轴向(Z 轴)进刀量, 取值范围 $0 < \Delta k \leq 9999999$ (单位: 最小输入增量, 无符号)。

R(Δd): 切削至径向切削终点后, 轴向(Z 轴)的退刀量, 取值范围 $0 \sim 99999999 \times$ 最小输入增量, (单位: mm/inch, 无符号)。省略R(Δd)时, 系统默认径向切削终点后, 轴向(Z 轴)的退刀量为0。省略Z(W)和Q(Δk), 默认往正方向退刀。



G75轨迹图

代码执行过程:

- ① 从径向切削循环起点 A_n 径向(X 轴)切削进给 Δi , 切削终点X 轴坐标小于起点X 轴坐标时, 向X 轴负向进给, 反之则向X 轴正向进给;
- ② 径向(X 轴)快速移动退刀 e , 退刀方向与①进给方向相反;
- ③ 如果X 轴再次切削进给($\Delta i+e$), 进给终点仍在径向切削循环起点 A_n 与径向进刀终点 B_n 之间, X 轴再次切削进给($\Delta i+e$), 然后执行②; 如果X 轴再次切削进给($\Delta i+e$)后, 进给终点到达 B_n 点或不在 A_n 与 B_n 之间, X 轴切削进给至 B_n 点, 然后执行④;
- ④ 轴向(Z 轴)快速移动退刀 Δd 至 C_n 点, B_f 点(切削终点)的Z 轴坐标小于A 点(起点)Z 轴坐标时, 向Z 轴正向退刀, 反之则向Z 轴负向退刀;
- ⑤ 径向(X 轴)快速移动退刀至 D_n 点, 第 n 次径向切削循环结束。如果当前不是最后一次径向切削循环, 执行⑥; 如果当前是最后一次径向切削循环, 执行⑦;
- ⑥ 轴向(Z 轴)快速移动进刀, 进刀方向与④退刀方向相反。如果Z 轴进刀($\Delta d+\Delta k$)后, 进刀终点仍在A 点与 A_f 点(最后一次径向切削循环起点)之间, Z 轴快速移动进刀($\Delta d+\Delta k$), 即: $D_n \rightarrow A_{n+1}$, 然后执行①(开始下一次径向切削循环); 如果Z 轴 进刀($\Delta d+\Delta k$)后, 进刀终点到达 A_f 点或不在 D_n 与 A_f 点之间, Z 轴快速移动至 A_f 点, 然后执行①, 开始最后一次径向切削循环;
- ⑦ Z 轴快速移动返回到起点A, G75 代码执行结束。

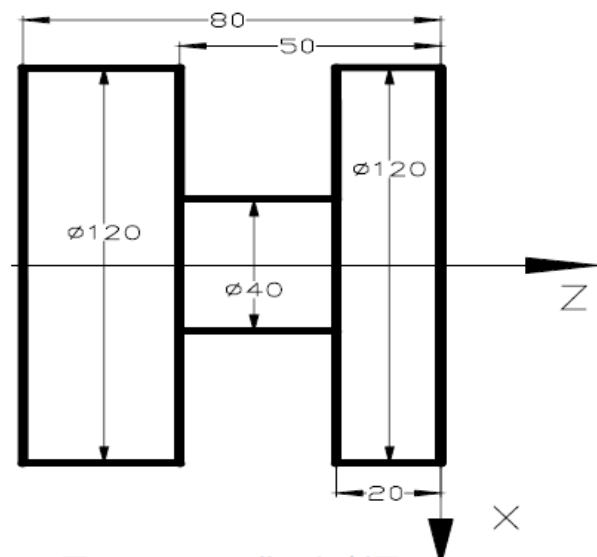
注意事项:

1. 循环动作是由含X(U)和P(Δi)的G75程序段进行的, 如果仅执行“G75 R(e);”程序段, 循环动作不进行;
2. Δd 和 e 均用同一地址R指定, 其区别是根据程序段中有无X(U)和P(Δi)代码字;

3. 在G75代码执行过程中, 可使自动运行停止并手动移动, 但要再次执行G75循环时, 必须返回到手动移动前的位置。如果不返回就再次执行, 后面的运行轨迹将错位;

4. 进行切槽循环时, 必须省略R(Δd)代码字, 因在切削至径向切削终点无退刀距离。

示例:



程序(假设切槽刀的宽度为4mm, 系统的最小增量为0.001mm):

O0008;

G00 X150 Z50 M3 S500; (启动主轴, 置转速500)

G0 X125 Z-24; (定位到加工起点, Z方向加上刀具宽度)

G75 R0.5 F150; (加工循环)

G75 X40 Z-50 P6000 Q3000; (X轴每次进刀6mm, 退刀0.5mm, 进给到终点(X40)后, 快速返回到起点(X125), Z轴进刀3mm, 循环以上步骤继续运行)

G0 X150 Z50; (返回到加工起点)

M30; (程序结束)

1.4.8 G76复合型螺纹切削固定循环

指令格式: G76 P(m)(r)(a) Q(Δd_{min}) R(d) J_ K_;

G76 X(U)_ Z(W)_ R(i) P(k)_ Q(Δd)_ F(I)_ ;

代码功能: 通过多次螺纹粗车、螺纹精车完成规定牙高(总切深)的螺纹加工, 如果定义的螺纹角度不为0°; 螺纹粗车的切入点由螺纹牙顶逐步移至螺纹牙底, 使得相邻两牙螺纹的夹角为规定的螺纹角度。G76 代码可加工带螺纹退尾的直螺纹和锥螺纹, 可实现单侧刀刃螺纹切削, 吃刀量逐渐减少, 有利于保护刀具、提高螺纹精度。G76 代码不能加工端面螺纹。

相关定义:

X: 螺纹终点X 轴绝对坐标;

U: 螺纹终点与起点X 轴绝对坐标的差值;

Z: 螺纹终点Z 轴的绝对坐标值;

W: 螺纹终点与起点Z 轴绝对坐标的差值;

P(m): 螺纹精车次数00~99(单位: 次), m 指定值执行后保持有效, 未输入m 时, 以系统数据参数043号的值作为精车次数。在螺纹精车时, 每次的进给的切削量等于螺纹精车的切削量。

P(r): 螺纹退尾长度00~99(单位: $0.1 \times L$, L 为螺纹螺距), r 指定值执行后保持有效, 未输入r 时, 以系统数据参数026号的值作为螺纹退尾宽度。螺纹退尾功能可实现无退刀槽的螺纹加工, 系统数据参数026号定义的螺纹退尾宽度对G92、G76 代码有效;

P(a): 相邻两牙螺纹的夹角, 取值范围为00~99, 单位: 度(°), a 指定值执行后保持有效, 未输入a 时, 以系统数据参数044号的值作为螺纹牙的角度。实际螺纹的角度由刀具角度决定, 因此a 应与刀具角度相同;

Q(Δd_{min}): 螺纹粗车时的最小切削量, 取值范围为0~99999(单位: 最小输入增量, 半径值)。当 $(\sqrt{n} - \sqrt{n-1}) \times \Delta d < \Delta d_{min}$ 时, 以 Δd_{min} 作为本次粗车的切削量, 即: 本次螺纹切深为 $(\sqrt{n-1} \times \Delta d + \Delta d_{min})$ 。设置 Δd_{min} 是为了避免由于螺纹粗车切削量递减造成粗车切削量过小、粗车次数过多。Q(Δd_{min})执行后, 指定值 Δd_{min} 保持有效, 未输入Q(Δd_{min})时, 以系统数据参数045号的值作为最小切削量;

R(d): 螺纹精车的切削量, 取值范围为00~99.999, (单位: mm/inch, 无符号, 半径值), 半径值等于螺纹精车切入点 B_e 与最后一次螺纹粗车切入点 B_f 的X 轴绝对坐标的差值。R(d) 执行后, 指定值d 保持有效, 未输入R(d)时, 以系统数据参数046的值作为螺纹精车切削量;

R(i): 螺纹锥度, 螺纹起点与螺纹终点X 轴绝对坐标的差值, 取值范围为-99999.999~99999.999(单位: mm/inch, 半径值)。未输入R(i)时, 系统按R(i)=0(直螺纹)处理;

P(k): 螺纹牙高, 螺纹总切削深度, 取值范围为1~99999999(单位: 最小输入增量, 半径值、无符号)。未输入P(k)时, 系统报警;

Q(Δd): 第一次螺纹切削深度, 取值范围为1~99999999(单位: 最小输入增量, 半径值、无符号)。未输入 Δd 时, 系统报警;

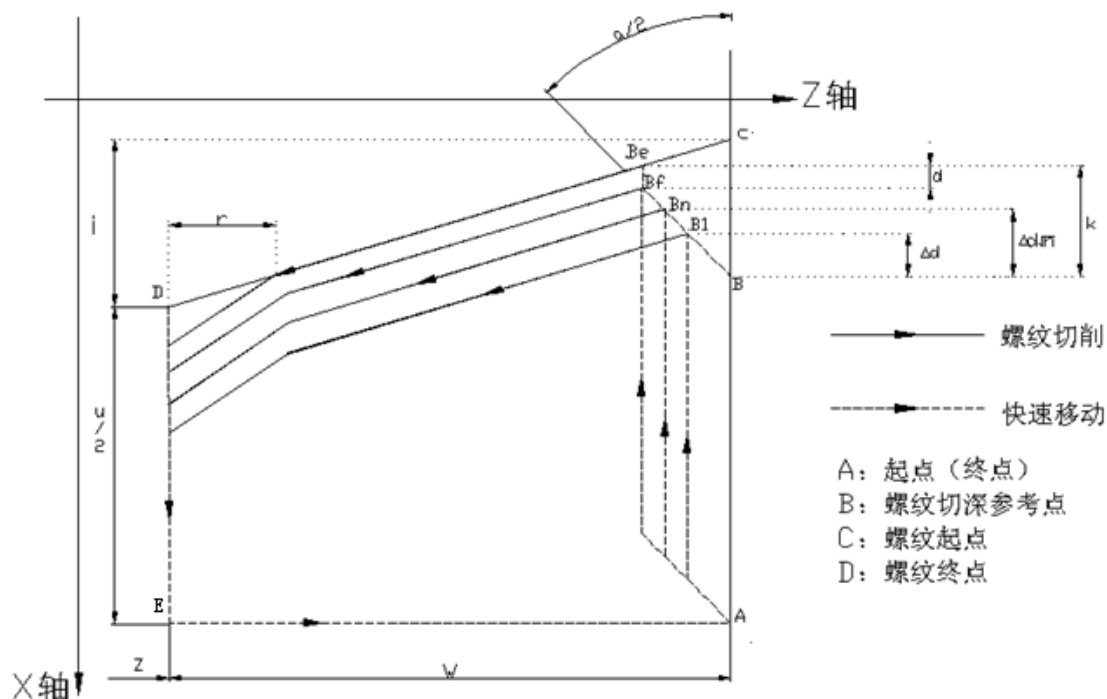
F: 螺纹导程;

I: 螺纹每英寸的螺纹牙数;

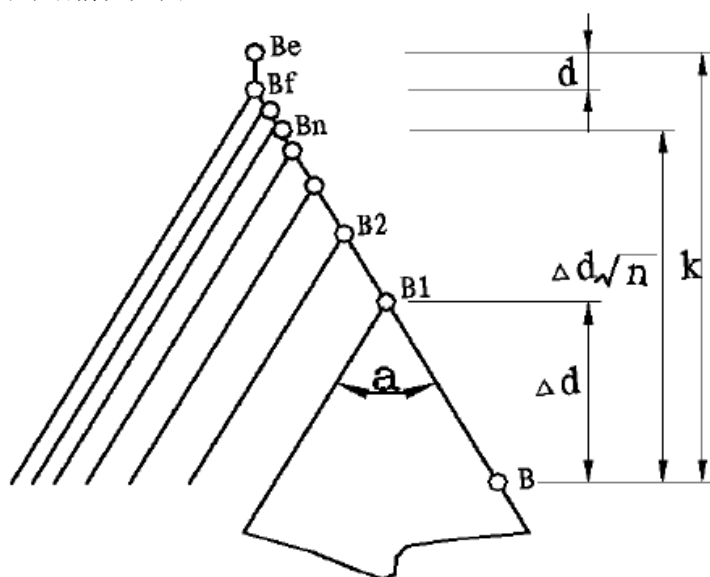
J: 螺纹退尾时在短轴方向的移动量(退尾量)(取值范围0~99999999×最小输入增量, 单位mm/inch, 不带方向); 如果短轴是X 轴, 该值为半径指定, 非模态参数;

K: 螺纹退尾时在长轴方向的长度(取值范围0~99999999×最小输入增量, 单位mm/inch, 不带方向); 如果长轴是X 轴, 则该值为半径指定, 非模态参数;

注：G76 没有编写J,K，退尾的使用方法和以前一样，按照P(r)或19 号参数退尾；若编写了J 或K 或J,K，退尾方式和G32、G92 相同；



切入方法的详细情况见图：



螺纹螺距指主轴转一圈长轴的位移量(X 轴位移量按半径值),C 点与D 点Z 轴坐标差的绝对值大于X 轴坐标差的绝对值(半径值, 等于i 的绝对值)时, Z 轴为长轴; 反之, X 轴为长轴。

代码执行过程:

- ①从起点快速移动到B₁, 螺纹切深为 Δd 。如果 $\alpha=0$, 仅移动X 轴; 如果 $\alpha \neq 0$, X 轴和Z 轴同时移动, 移动方向与A→D 的方向相同;
- ②沿平行于C→D 的方向螺纹切削到与D→E 相交处($r \neq 0$ 时有退尾过程);
- ③X 轴快速移动到E 点;

④Z 轴快速移动到A 点，单次粗车循环完成；

⑤再次快速移动进刀到 B_n (n 为粗车次数)，切深取 $(\sqrt{n} \times \Delta d)$ 、 $(\sqrt{n-1} \times \Delta d + \Delta d_{\min})$ 中的较大值，如果切深小于 $(k-d)$ ，转②执行；如果切深大于或等于 $(k-d)$ ，按切深 $(k-d)$ 进刀到 B_f 点，转⑥执行最后一次螺纹粗车；

⑥ 沿平行于 $C \rightarrow D$ 的方向螺纹切削到与 $D \rightarrow E$ 相交处($r \neq 0$ 时有退尾过程)；

⑦ X 轴快速移动到E 点；

⑧ Z 轴快速移动到A 点，螺纹粗车循环完成，开始螺纹精车；

⑨快速移动到 B_e 点(螺纹切深为 k 、切削量为 d)后，进行螺纹精车，最后返回A 点，完成一次螺纹精车循环；

⑩如果精车循环次数小于 m ，转⑨进行下一次精车循环，螺纹切深仍为 k ，切削量为0；如果精车循环次数等于 m ，G76 复合螺纹加工循环结束。

注意事项：

1. 螺纹切削过程中执行进给保持操作后，系统仍进行螺纹切削，螺纹切削完毕，显示“系统暂停”，程序运行暂停；

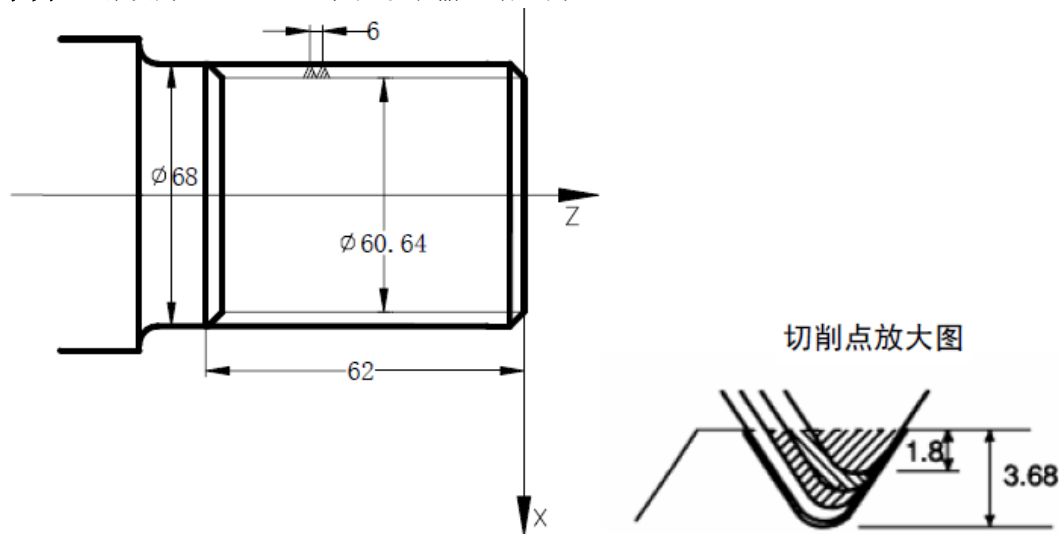
2. 系统复位、急停或驱动报警时，螺纹切削减速停止；

3. G76 P(m)(r)(a) Q($\Delta d_{\min}$) R(d)可全部省略或省略部分代码地址，省略的地址按参数设定值运行；

4. m 、 r 、 a 用同一个代码地址P一次输入， m 、 r 、 a 全部省略时，按数据参数043、026、044号设定值运行；地址P输入1位或2位数时取值为 a ；地址P输入3位或4位数时取值为 r 与 a ；

5. U、W的符号决定了 $A \rightarrow C \rightarrow D \rightarrow E$ 的方向，R(i)的符号决定了 $C \rightarrow D$ 的方向。U、W的符号有四种组合方式，对应四种加工轨迹，参考G92代码。

示例： 螺纹为M68×6。（系统最小输入增量为0.0001mm）



程序：O0013；

G50 X100 Z50 M3 S300；	(设置工件坐标系启动主轴，指定转速)
G04 X2；	(延时2S，主轴转速稳定)
G00 X80 Z10；	(快速移动到加工起点)
G76 P020560 Q150 R0.1；	(精加工重复次数2，倒角宽度0.5mm,刀具角度60°； 最小切入深度0.15,精车余量0.1)
G76 X60.64 Z-62 P3680 Q1800 F6；	(螺纹牙高3.68，第一螺纹切削深度1.8)
G00 X100 Z50 ；	(返回程序起点)
M30；	(程序结束)

1.4.9 G80~G89 钻孔固定循环

项目	钻孔方向	孔底位置动作	逃离动作	用途
G80钻孔循环取消	-----	-----	-----	取消循环
G82端面钻孔、点钻循环	Z	暂停	快速进给	点钻循环
G83端面排屑钻孔循环	Z	暂停	快速进给	钻孔循环
G84端面刚性攻牙循环	Z	钻头反钻	切削进给	攻牙循环
G85端面搪孔循环	Z	暂停	切削进给	搪孔循环
G86侧面钻孔、点钻循环	X	暂停	快速进给	点钻循环
G87侧面排屑钻孔循环	X	暂停	快速进给	钻孔循环
G88侧面刚性攻牙循环	X	钻头反钻	切削进给	攻牙循环
G89侧面搪孔循环	X	暂停	切削进给	搪孔循环

1.4.10 G82、G86钻孔、点钻循环

指令格式： G82 X(U)_ Z(W)_ R_ Q_ P_ F_ K_ M_；（端面钻孔）

G86 X(U)_ Z(W)_ R_ Q_ P_ F_ K_ M_；（侧面钻孔）

代码说明：

X(U)：洞孔位置的坐标数据；--- G82

Z(W)：孔底位置绝对值（从R点到洞底的增加值）；--- G82

Z(W)：洞孔位置的坐标数据；--- G86

X(U)：孔底位置绝对值（从R点到洞底的增加值）；--- G86

R：R点位置的绝对值；

Q：每次进给深度（正负号无效）；

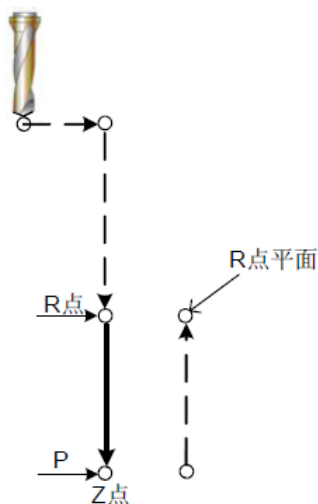
P：洞底暂停时间（毫秒）；

F：进给速率；

K：重复次数；

M：C轴夹紧M码, M+1为C轴松开；

1、当代码没有指定Q时，用作正常钻孔切削进给，执行到孔底，然后刀具从孔底快速移动退回，如图：



循环过程：

- ① 快速定位到洞孔位置；
- ② 如果指令了M代码，执行相应的M功能；
- ③ 快速定位到R点位置；
- ④ 以切削进给速度钻孔到孔底位置；
- ⑤ 如果指令了P代码，则在孔底位置暂停P指定的时间；
- ⑥ 快速退回到R点位置；
- ⑦ 如果指令了M代码，则执行相应的M+1功能，否则循环结束。

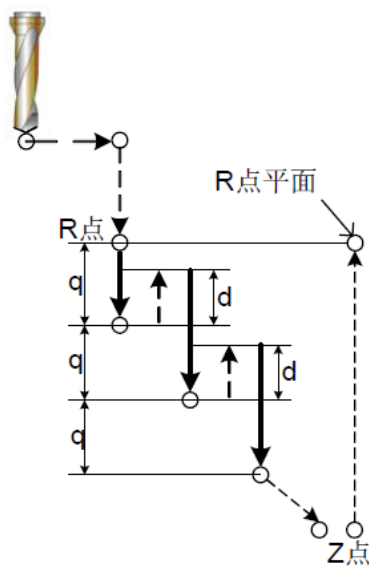
示例：（一般钻孔，假设M51为C轴夹紧命令，M52为C轴松开命令）

```

M03 S1000;          （主轴正转1000rpm）
G00 X50;            （快速定位至起始点）
G82 Z-40.0 C0.0 R2.0 P1000 F100 M51; （C轴0度钻第一个孔）
C90.0 M51;          （C轴90度钻第二个孔）
C180.0 M51;         （C轴180度钻第三个孔）
G80;                （取消钻孔循环）
M30;

```

2、当代码有指定Q时，专门为执行高速深孔钻设定，它执行间歇切削进给直到孔的底部，在进给的同时从孔中快速退刀，排出切屑。如图：



循环过程：

- ① 快速定位到洞孔位置；
- ② 如果指令了M代码，执行相应的M功能；
- ③ 快速定位到R点位置；
- ④ 切削进给Q距离；
- ⑤ 快速退刀d距离（退刀距离d由数据参数089号设定）；
- ⑥ 切削进给(Q+d)距离；
- ⑦ 循环⑤⑥直至到达加工孔底；
- ⑧ 如果指令了P代码，则在孔底位置暂停P指定的时间；

- ⑨ 快速退回到 R 点位置;
- ⑩ 如果指令了 M 代码, 则执行相应的 M+1 功能, 否则循环结束。

示例: 端面高速深孔钻, 假设 M51 为 C 轴夹紧命令, M52 为 C 轴松开命令

M03 S1000;	(主轴转速1000rpm)
G00 X50;	(快速定位到起始点)
G82 Z-40.0 C0.0 R2.0 P1000 Q5000 F100 M51;	(C轴0度钻第一孔, 每次钻深5mm)
C90 M51;	(C轴90度钻第二孔)
C180 M51;	(C轴180度钻第三孔)
G80;	(取消循环)
M30;	(程序结束)

1.4.11 G83、G87排屑钻孔循环

指令格式： G83 X(U)_ Z(W)_ R_ Q_ P_ F_ K_ M_；（端面排屑钻孔）
G87 X(U)_ Z(W)_ R_ Q_ P_ F_ K_ M_；（侧面排屑钻孔）

代码说明：

X(U)：洞孔位置的坐标数据；--- G83

Z(W)：孔底位置绝对值（从R点到洞底的增加值）；--- G83

Z(W)：洞孔位置的坐标数据；--- G87

X(U)：孔底位置绝对值（从R点到洞底的增加值）；--- G87

R：R点位置的绝对值；

Q：每次进给深度（正负号无效）；

P：洞底暂停时间（毫秒）；

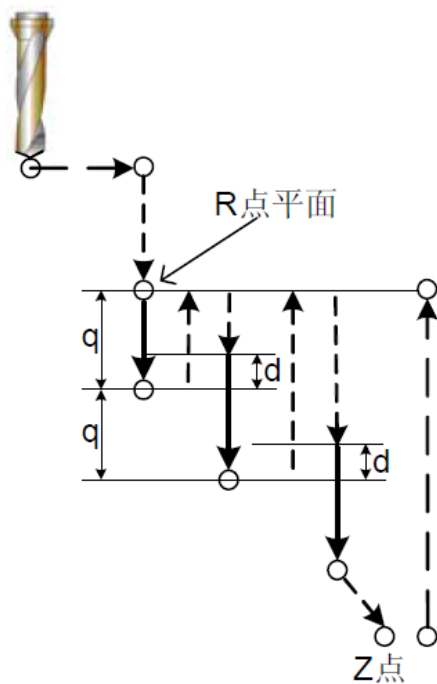
F：进给速率；

K：重复次数；

M：C轴夹紧M码, M+1为C轴松开；

循环过程：

- ①快速定位到洞孔位置；
- ②如果指令了M代码，执行相应的M功能；
- ③快速定位到R点位置；
- ④切削进给Q距离；
- ⑤快速退回至R点位置；
- ⑥快速进刀至离末加工面d距离处（d由数据参数089号设定）；
- ⑦切削进给(Q+d)距离；
- ⑧循环⑤⑥⑦直至到达加工孔底；
- ⑨如果指令了P代码，则在孔底位置暂停P指定的时间；
- ⑩快速退回到R点位置；
- ⑪如果指令了M代码，则执行相应的M+1功能，否则循环结束。



示例：端面高速排屑钻深孔，假设M51为C轴夹紧命令，M52为C轴松开命令

M03 S1000;	(主轴转速1000rpm)
G00 X50;	(快速定位到起始点)
G83 Z-40.0 C0.0 R2.0 P1000 Q5000 F100 M51;	(C轴0度钻第一孔，每次钻深5mm)
C90 M51;	(C轴90度钻第二孔)
C180 M51;	(C轴180度钻第三孔)
G80;	(取消循环)
M30;	(程序结束)

1.4.12 G84、G88刚性攻牙循环

指令格式： G84 X(U)_ Z(W)_ P_ F(I)_ M_；（端面刚性攻牙）

G88 X(U)_ Z(W)_ P_ F(I)_ M_；（侧面刚性攻牙）

代码说明：

X(U)：攻牙洞孔位置的坐标数据；—— G84

Z(W)：攻牙孔底位置绝对值；—— G84

Z(W)：攻牙洞孔位置的坐标数据；—— G88

X(U)：攻牙孔底位置绝对值；—— G88

P：洞底暂停时间（毫秒）；

F(I)：螺纹的导程，F(I) > 0 右旋攻丝，F(I) < 0 左旋攻丝；

M：C轴夹紧M码，M+1为C轴松开；

指定刚性攻牙的方法：

在 G84/G88 指令之前指定M29 S_，S为攻牙时的转速，如下：

M29 S_；

G84(G88) X_C_(Z_C_) Z_(X_) P_ F_ M_；

循环过程：

①动作1，快速定位到攻牙孔位置；

②在开始动作 2 之前，如果有用于夹紧主轴的M 代码，则输出M α ；

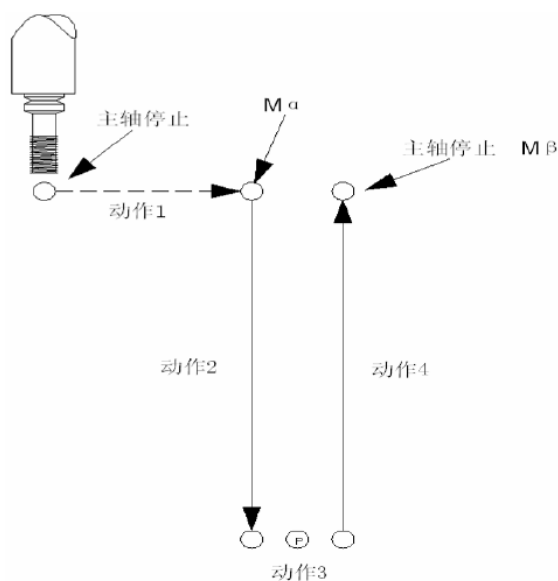
③动作2，刚性攻牙开始；

④动作3，攻牙至孔底暂停时间P；

⑤动作4，刚性攻牙回退至起点；

⑥如果刚性攻丝程序段指定了夹紧主轴的 M 代码，则此处输出M β ；

注： α 值在M指令中设定， $\beta = \alpha + 1$ 。



示例：假设M51为C轴夹紧命令，M52为C轴松开命令

M05;	(主轴停止)
G00 Z3;	(快速定位到Z轴起始点)
M29 S500;	(指令刚性攻牙，转速500rpm)
G84 X10 C0 Z-20 P1000 F1.0 M51;	(C轴0度攻第一孔)
C90 M51;	(C轴90度攻第二孔)
C180 M51;	(C轴180度攻第三孔)
G80;	(取消攻牙循环)
G00 Z50;	
M30;	(程序结束)

1.4.13 G85、G89搪孔循环

指令格式： G85 X(U)_ Z(W)_ R_ P_ F_ K_ M_；（端面搪孔）
 G89 X(U)_ Z(W)_ R_ P_ F_ K_ M_；（侧面搪孔）

代码说明：

X(U)：洞孔位置的坐标数据；--- G85

Z(W)：孔底位置绝对值（从R点到洞底的增加值）；--- G85

Z(W)：洞孔位置的坐标数据；--- G89

X(U)：孔底位置绝对值（从R点到洞底的增加值）；--- G89

R：R点位置的绝对值；

P：洞底暂停时间（毫秒）；

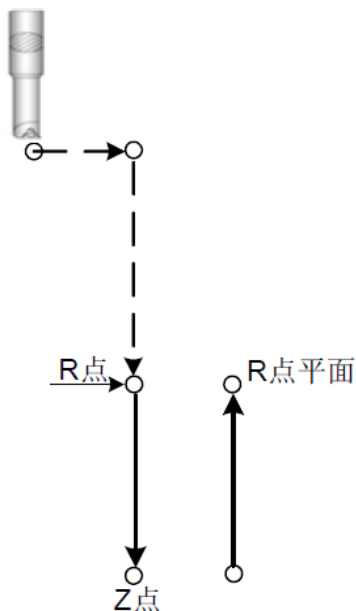
F：进给速率；

K：重复次数；

M：C轴夹紧M码, M+1为C轴松开；

循环过程：

- ①快速定位到洞孔位置；
- ②如果指令了M代码，执行相应的M功能；
- ③快速定位到R点位置；
- ④切削进给至孔底位置；
- ⑤如果指令了P代码，则在孔底位置暂停P指定的时间；
- ⑥切削进给至R点位置；
- ⑧ 如果指令了M代码，则执行相应的M+1功能，否则循环结束。



示例：

M03 S1000； （主轴转速1000rpm）
 G00 X50； （快速定位到起始点）
 G85 Z-40 C0.0 R2.0 P1000 F100 M51（C轴0度搪第1孔）
 C90 M51； （C轴90度搪第2孔）
 C180 M51； （C轴180度搪第3孔）
 G80； （取消循环）
 M30； （程序结束）

1.5.0 G50工件坐标系设定

指令格式: G50 X(U)_ Z(W)_ ;

代码功能: 设置当前位置的绝对坐标, 通过设置当前位置的绝对坐标在系统中建立工件坐标系(也称浮动坐标系)。工件坐标系建立后, 绝对坐标编程按这个坐标系输入坐标值, 直至再次执行G50 建立新的工件坐标系。

代码说明:

G50 为非模态G 代码;

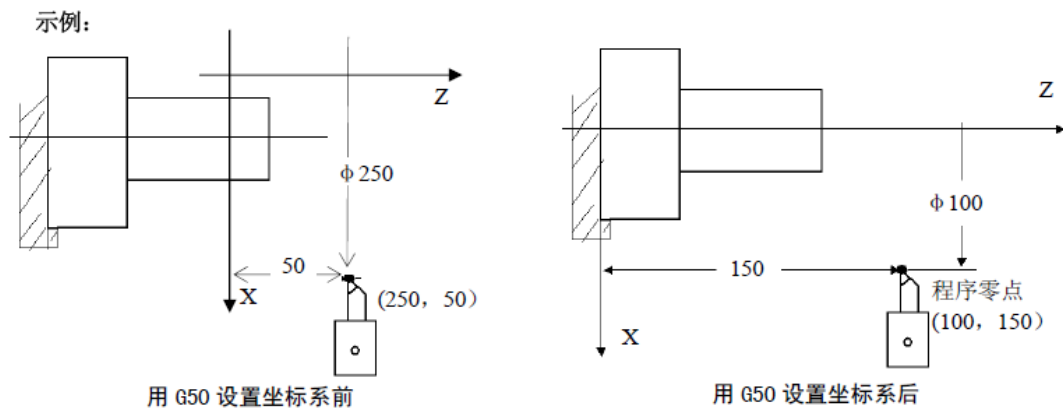
X: 当前位置新的X 轴绝对坐标;

U: 当前位置新的X 轴绝对坐标与执行代码前的绝对坐标的差值;

Z: 当前位置新的Z 轴绝对坐标;

W: 当前位置新的Z 轴绝对坐标与执行代码前的绝对坐标的差值;

G50 代码中, X(U)、Z(W)均未输入时, 不改变当前坐标值。未输入X(U)或Z(W), 未输入的坐标轴保持原来设定的程序零点(当G50 SXXXX 时不设置工件坐标系)。



当执行代码段“G50 X100 Z150;”后, 建立了如图所示的工件坐标系。

1.5.1 恒线速控制G96、恒转速控制G97

指令格式：G96 S__；（S0000～S9999，前导零可省略）

代码功能：恒线速控制有效、给定切削线速度（m/min），取消恒转速控制。G96 为模态G 代码，如果当前为G96 模态，可以不输入G96。

代码格式：G97 S__；（S0000～S9999，前导零可省略）

代码功能：取消恒线速控制、恒转速控制有效，给定主轴转速（r/min）。G97 为模态G 代码，如果当前为G97 模态，可以不输入G97。

代码格式：G50 S__；（S0000～S9999，前导零可省略）

代码功能：设置恒线速控制时的主轴最高转速限制值（r/min）。

G96、G97 为同组的模态代码字，只能一个有效。G97 为初态代码字，CNC 上电时默认G97 有效。

车床车削工件时，工件通常以主轴轴线为中心线进行旋转，刀具切削工件的切削点可以看成围绕主轴轴线作圆周运动，圆周切线方向的瞬时速率称为切削线速度（通常简称线速度）。不同材料的工件、不同材料的刀具要求的线速度不同。

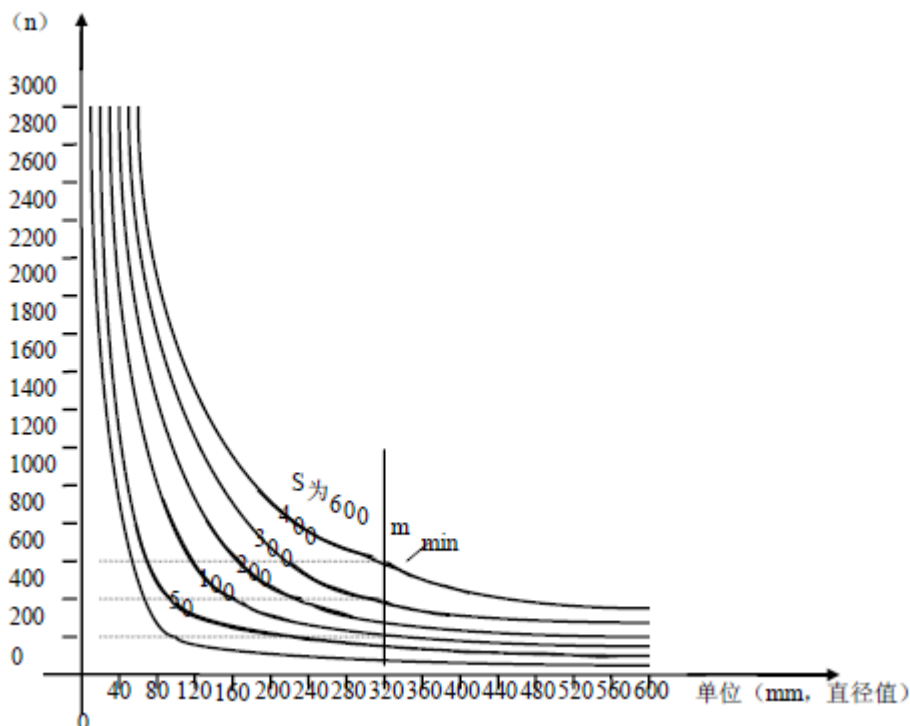
主轴转速模拟电压控制功能有效时，恒线速控制功能才有效。在恒线速控制时，主轴转速随着编程轨迹（忽略刀具长度补偿）的X 轴绝对坐标值的绝对值的变化，X 轴绝对坐标值的绝对值增大，主轴转速降低，X 轴绝对坐标值的绝对值减小，主轴转速提高，使得切削线速度保持为S 代码值。使用恒线速控制功能切削工件，可以使得直径变化的工件表面光洁度保持一致。

线速度=主轴转速 × |X| × π ÷1000 （m/min）

主轴转速：r/min

|X|：X 轴绝对坐标值的绝对值（直径值），mm

$\pi \approx 3.14$



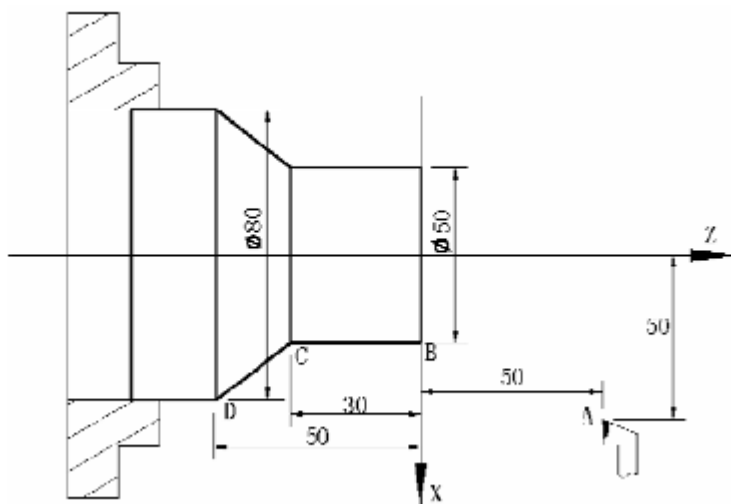
恒线速控制时，只在切削进给（插补）过程中随着编程轨迹X 轴绝对坐标值的绝对值的

变化改变主轴转速，对于G00 快速移动，由于不进行实际切削，G00 执行过程中主轴转速保持不变，此时的主轴转速按程序段终点位置的线速度计算。

恒线速控制时，工件坐标系的Z 坐标轴必须与主轴轴线（工件旋转轴）重合，否则，实际线速度将与给定的线速度不一致。

恒线速控制有效时，G50 S__可限制主轴最高转速（r/min），当按线速度和X 轴坐标值计算的主轴转速高于G50 S__设置的这个限制主轴最高转速限制值时，实际主轴转速为主轴最高转速限制值。CNC 上电时，主轴最高转速限制值未设定、主轴最高转速限制功能无效。G50 S__定义的最高转速限制值在重新指定前是保持的，最高转速限制功能在G96 状态下有效，在G97 状态下G50 S__设置的主轴最高转速不起限制作用，但主轴最高转速限制值仍然保持。

示例：



```

00001 ;           (程序名)
N0010 M3 G96 S300; (旋转主轴、恒线速控制有效、线速度为300m/min)
N0020 G0 X100 Z50; (快速移动至A 点，移动过程中主轴转速为955r/min)
N0030 G0 X50 Z0;   (快速移动至B 点，移动过程中主轴转速为1910r/min)
N0040 G1 W-30 F200; (从B 点切削至C 点，切削中主轴转速恒为1910r/min)
N0050 X80 W-20 F150; (从C 点切削至D 点，主轴转速从1910r/min 线性变化为1194r/min)
N0060 G0 X100 Z50; (快速退回A 点，移动过程中主轴转速为955r/min)
N0110 M30;         (程序结束，关主轴、冷却液)
N0120 %

```

1.5.2 G98/G99（每分钟进给每转进给）

指令格式：：G98 F__；（给定每分进给速度）

代码功能：以mm/min 为单位给定切削进给速度，G98 为模态G 代码，如果当前为G98 模态，可以不输入G98。

指令格式：G99 F__；

代码功能：以毫米/转为单位给定切削进给速度，G99 为模态G 代码。，如果当前为G99 模态，可以不输入G99。CNC 执行G99 F__时，把F 代码值（毫米/转）与当前主轴转速（r/min）的乘积作为代码进给速度控制实际的切削进给速度，主轴转速变化时，实际的切削进给速度随着改变。使用G99 F__给定主轴每转的切削进给量，可以在工件表面形成均匀的切削纹路。在G99 模态进行加工，机床必须安装主轴编码器。

G98、G99 为同组的模态G 代码，只能一个有效。G98 为初态G 代码，CNC 上电时默认G98 有效。

每转进给量与每分钟进给量的换算公式：

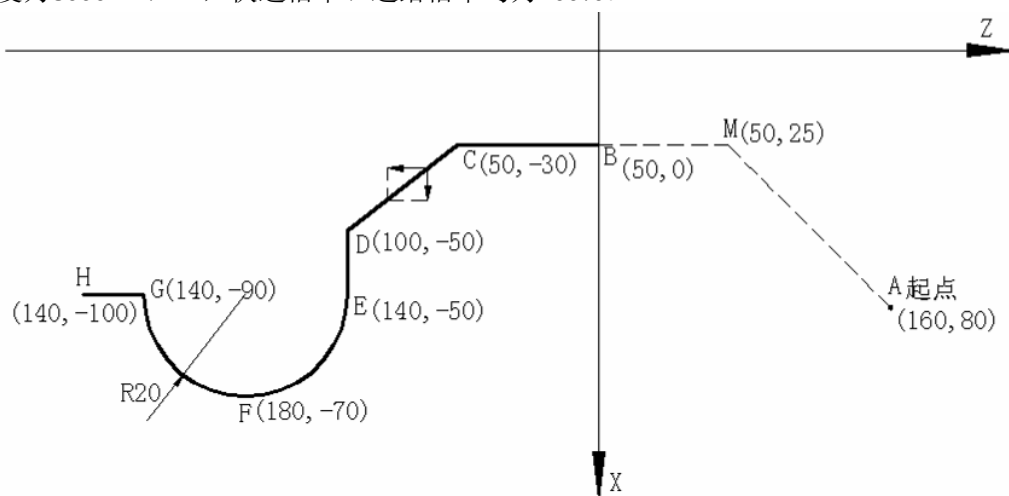
$$F_m = F_r \times S$$

其中：F_m：每分钟的进给量（mm/min）；

F_r：每转进给量（mm/r）；

S：主轴转速（r/min）。

示例：括号内为各点的坐标值（X轴为直径值），设置X轴速度为8000mm/min、Z轴速度为8000mm/min，快速倍率、进给倍率均为100%。



G50 X160 Z80; （建立工件坐标系）

G0 G98 X50 Z0; （从A点经M点快速移动至B点。A→M中，X轴速度为8000mm/min,Z轴速度为8000mm/min, M→B中，X轴速度为0mm/min, Z轴速度为8000mm/min）

G1 W-30 F100; （B→C，整个过程中X轴速度为0mm/min，Z轴速度为100mm/min）

X100 W-20; （C→D，整个过程中X轴速度为156mm/min，Z轴速度为62mm/min）

X140; （D→E，整个过程中X轴速度为200mm/min，Z轴速度为0mm/min）

G3 W-40 R20; （EFG圆弧插补，E点X轴速度为200mm/min，Z轴速度为0mm/min
F点X轴速度为0mm/min，Z轴速度为100mm/min G点X轴速度为
200mm/min，Z轴速度为0mm/min）

W-10; （G→H，整个过程中X轴速度为0mm/min，Z轴速度为100mm/min）

M30;

1.5.3 G91附加轴指定增量方式

指令格式：G91

代码说明：作为指令轴移动量的方法，有绝对值指令和增量值指令两种方法。绝对值指令是用轴移动的终点位置的坐标值进行编程的方法。增量值指令是用轴移动量直接编程的方法。

下表是各轴使用绝对值指令和增量值指令时的编程方法：

	绝对值指令	增量值指令
X轴	X__	U__
Y轴	Y__	V__
Z轴	Z__	W__
A轴	A__	G91 A__
B轴	B__	G91 B__
C轴	C__	G91 C__

示例：C轴当前位置为30度，需要转到180度的位置：

绝对值指令时的方法：

G00 C180;

增量值指令时的方法：

G00 G91 C150;

2、M代码指令说明

M码	功能
M00	程序暂停
M01	选择性程序暂停
M02	程序结束，程序光标不返回开头
M03	主轴正转
M04	主轴反转
M05	主轴停止
M08	冷却水泵开
M09	冷却水泵关
M10	夹头夹紧
M11	夹头松开
M12	尾座进
M13	尾座退
M14	主轴准停定向打开
M15	主轴准停定向关闭
M16	主轴切换到位置模式
M17	主轴切换到速度模式
M18	工件计数加1
M23	第二模拟量输出或A轴脉冲速度旋转正转
M24	第二模拟量输出或A轴脉冲速度旋转反转
M25	A轴脉冲速度旋转停止
M30	程序结束，程序光标返回开头
M32	润滑油泵开
M33	润滑油泵关
M35	自动重复打料功能
M41	档位主轴第一档
M42	档位主轴第二档
M43	档位主轴第三档
M44	档位主轴第四档
M51	送料气缸1开
M52	送料气缸1关
M53	送料气缸2开
M54	送料气缸2关
M55	送料气缸3开
M56	送料气缸3关
M57	送料气缸4开
M58	送料气缸4关
M59	打料气缸开
M60	打料气缸关

M61	送料气缸5开
M62	送料气缸5关
M63	送料气缸6开
M64	送料气缸6关
M65	送料气缸7开
M66	送料气缸7关
M89	检测常开触点信号
M90	检测常闭触点信号
M96	检测通道状态，仅双通道系统有此功能
M97	启动通道运行，仅双通道系统有此功能
M98	呼叫子程序
M99	主程序循环或子程序返回主程序

2.1.0 M89, M90检测输入触点功能

1. 检测常开输入信号，PLC参数DT022设置检测不到信号报警时间：

指令格式：M89 Kx.x

代码功能：Kx.x代表要检测的端口号，比如M89 K2.1 表示检测X2.1端口，当该端口输入信号接通时程序继续往下执行，否则一直等待，设定的时间未检测到信号PLC报警。

2. 检测常闭输入信号

指令格式：M90 Kx.x

代码功能：Kx.x代表要检测的端口号，比如M90 K2.1 表示检测X2.1端口，当该端口输入信号断开时程序继续往下执行，否则一直等待，设定的时间未检测到信号PLC报警。

2.1.1 双通道系统专用M代码

1、M96：通道 1 检测通道 2 是否处于待机状态

例：

M96 Q2

Q2 代表要检测通道 2。当通道 2 处于运行状态时，通道 1 就一直等待，直至通道 2 处于待机状态时，通道 1 才继续往下执行。

2、M97 Pxxxx：通道1启动通道2，通道2运行程序Oxxxx号

P 代表通道 2 要运行的程序号，执行该指令时，通道 1 会自动检测通道 2 的运行状态，如果通道 2 处于运行状态，通道 1 会跳过该指令继续往下执行。如果通道 2 处于待机状态，通道 2 就立即执行 P 指定的程序，同时通道 1 继续往下执行。

当要调用的程序号不存在时，系统报警。

按下进给保持键时，双通道同时暂停，再次按下循环启动键时，双通道同时运行。

例：通道1启动通道2运行，通道2运行O0005号程序：

M97 P0005;

3、通道间互相等待的M代码（M100～M199）

例： 通道 1 程序

G0 X0 Y0;

M100;

G1 X50;

.....

通道 2 程序

G0 X0 Y100;

G1 X100;

M100;

.....

通道 1 执行到 M100 代码时，就进入等待通道 2 的状态。直到通道 2 也执行到 M100 代码时，那么通道 1 与通道 2 就继续往下执行各自的程序。

通道 1 和通道 2 哪个通道先执行到对应的 M 代码时就等待另一个通道也执行了相应的 M 代码后才继续往下执行。

M100~M199可任意使用。但是两通道间使用的M代码必须一一对应。