

Contents

1. 快速定位.....	3
2. 直线插补.....	3
3. 圆弧插补.....	3
3.1 终点圆心式.....	3
3.2 终点半径式.....	4
4. 螺旋线插补	4
5. 空间三点圆弧插补.....	5
6. 简化轨迹计算.....	6
6.1 自动倒角.....	6
6.2 自动倒圆.....	6
6.3 角度编程.....	7
6.4 平面相切圆弧	7
7. 极坐标运动指令	8
8. 作者信息	9

1. 快速定位

快速定位指令 **G00** 用于在非切削状态下的刀具定位。该速度值不受 **F** 值的约束，将以各轴最快的速度运行。

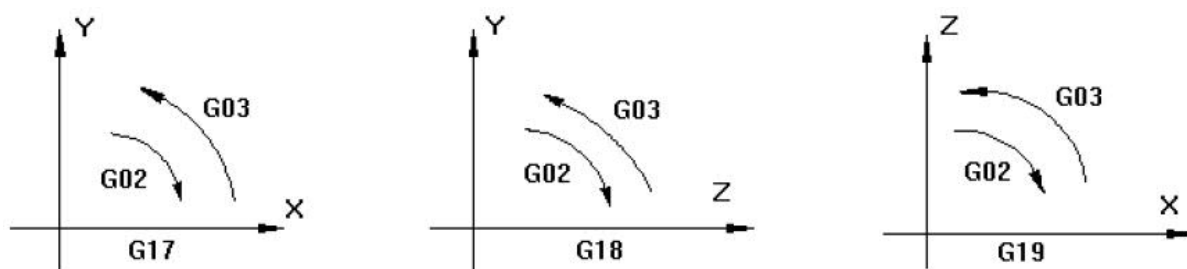
虽然快速定位的速度并没有指令代码可以控制，但是定位的过程仍然遵循插补的方式。也就是说，在快速定位的过程中刀具在空间的运动轨迹仍然是一条直线段。

2. 直线插补

直线插补的指令代码为 **G01**，运行特点是两点之间的轨迹保持为一条直线段。同时，插补的速度可以用 **F** 指令进行控制。

3. 圆弧插补

圆弧插补是一种在基本加工平面内的圆弧轨迹，因此在进行圆弧插补之前要先用 **G17 / G18 / G19** 指令选择适合的加工平面。另外，圆弧插补具有严格的方向性，**G02** 为顺时针圆弧插补，**G03** 为逆时针圆弧插补。顺、逆时针方向的判定请参照下图。



西门子系统的圆弧插补形式多样，下面以最常用的 **G17** 平面为例，介绍两种常用的形式：

3.1 终点圆心式

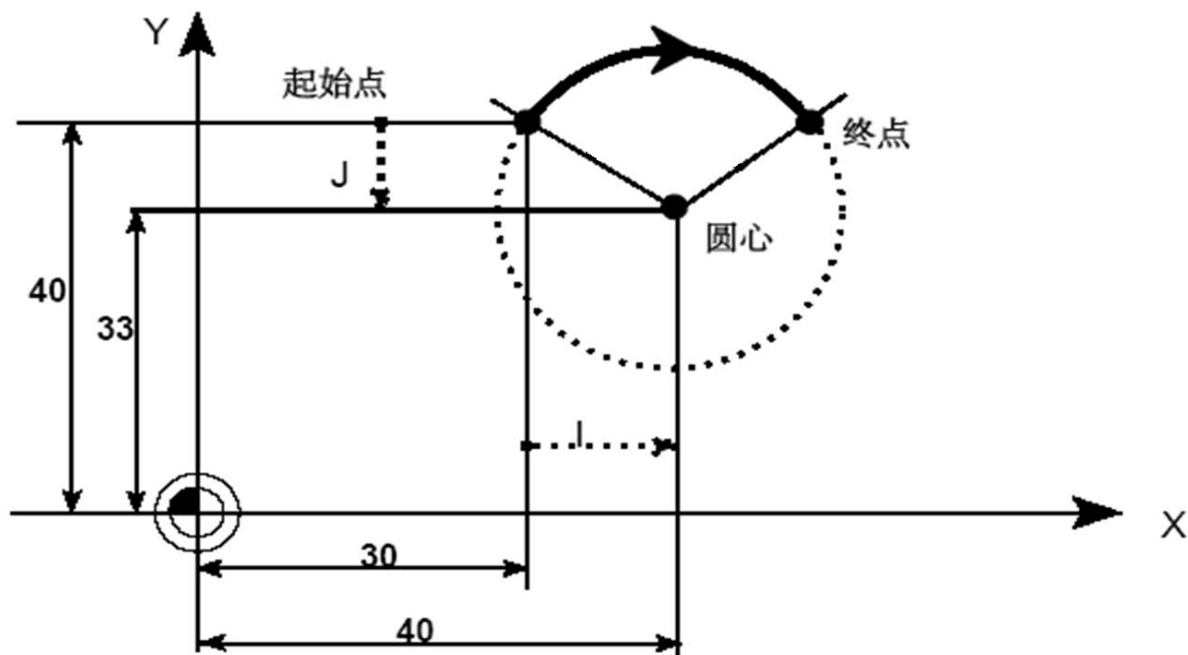
指令形式：**G02/G03 X... Y... I... J...**

X... Y...：圆弧的终点坐标

I... J...或者 **I=AC(...) J=AC(...)**：圆心的坐标

一般情况下，**I...**表示圆心相对于起点在 **X** 轴方向的距离，**J...**表示圆心相对于起点在 **Y** 轴方向的距离。

编程示例：圆弧起点坐标为 **X30 Y40**，圆弧终点坐标为 **X50 Y40**，圆心如下图所示。



圆弧插补指令应为：G02 X50 Y40 I10 J-7

但是，如果结合绝对坐标的非模态指令 AC 一起使用，I、J 就可以表示圆心的绝对坐标了。
I=AC(...)表示圆心坐标在 X 轴上的绝对坐标，J=AC(...)表示圆心在 Y 轴上的绝对坐标。则上述圆弧插补指令也可以写做：G02 X50 Y40 I=AC(40) J=AC(33)

3.2 终点半径式

指令形式：G02/G03 X... Y... CR=...

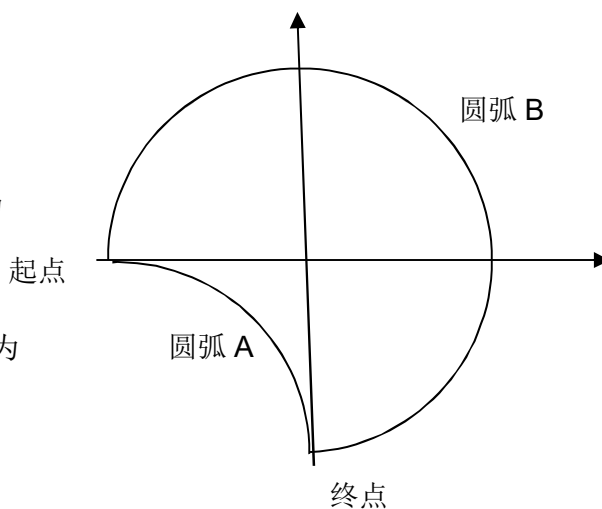
X... Y...：圆弧的终点坐标

CR=...：数值表示圆弧半径，正号表示轨迹为小圆弧，负号表示轨迹为大圆弧。

如右图所示，圆弧起点为 X-15 Y0，圆弧终点为 X0 Y-15，圆弧半径为 15。

圆弧 A 的指令应为：G2 X0 Y-15 CR=15

圆弧 B 的指令应为：G2 X0 Y-15 CR= -15



4. 螺旋线插补

螺旋线插补的指令形式及要求与圆弧插补基本相同，仍然受到基本加工平面的限制，只是将平面内的圆弧线段沿着刀具轴线的方向叠加了一个拉伸运动。

例如：

G02 X50 Y40 Z--11 I10 J-7 TURN=1

G02 X50 Y40 Z-11 I=AC(40) J=AC(33) TURN=1

G02 X0 Y-15 Z-5 CR=15 TURN=1

G02 X0 Y-15 Z-3 CR= -15 TURN=0

Z... : 圆弧终点在垂直于加工平面（G17）的第三轴（Z 轴）上的坐标值

TURN=... : 圆弧在加工平面内包含整圆的圈数，或者说圆弧终点在加工平面内经过圆弧起点的次数

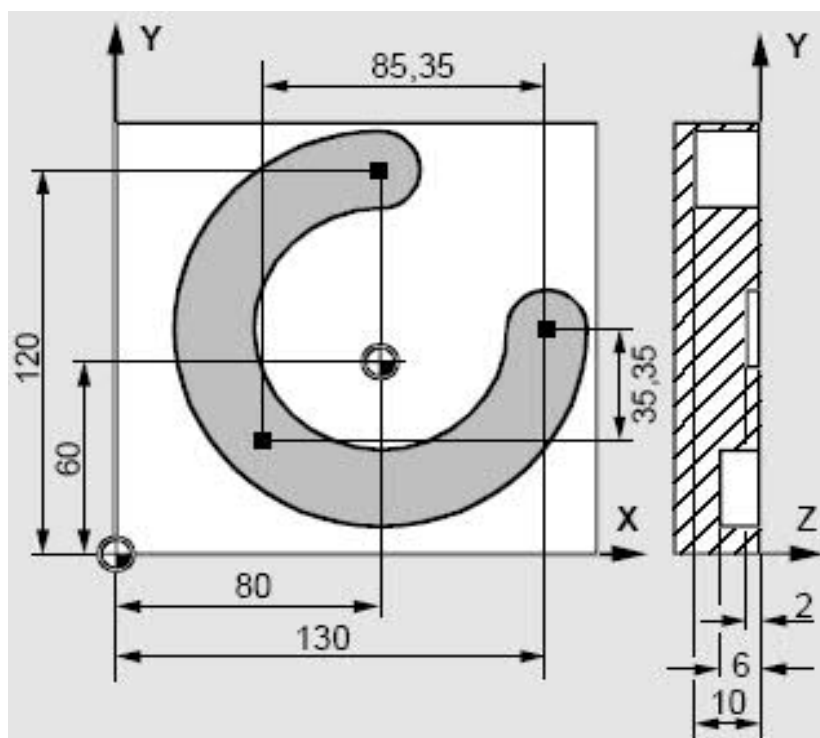
5. 空间三点圆弧插补

与前面基于平面的圆弧插补不同，CIP 圆弧插补可以不受基准平面（G17、G18、G19）的约束。可以在空间任意平面内走出圆弧的轨迹。

指令形式：CIP X... Y... Z... I1=... J1=... K1=...

X... Y... Z... : 圆弧终点的空间坐标

I1=... J1=... K1=... : 圆弧中间点的空间坐标



N10 G0 G90 X130 Y60 S800 M3 ; 定位到空间圆弧的起始点

N20 G17 G1 Z-2 F100 ; 刀具沿轴线方向进给向下切入工件

N30 CIP X80 Y120 Z-10 I1= IC(-85.35) J1=IC(-35.35) K1=-6 ; 空间圆弧插补

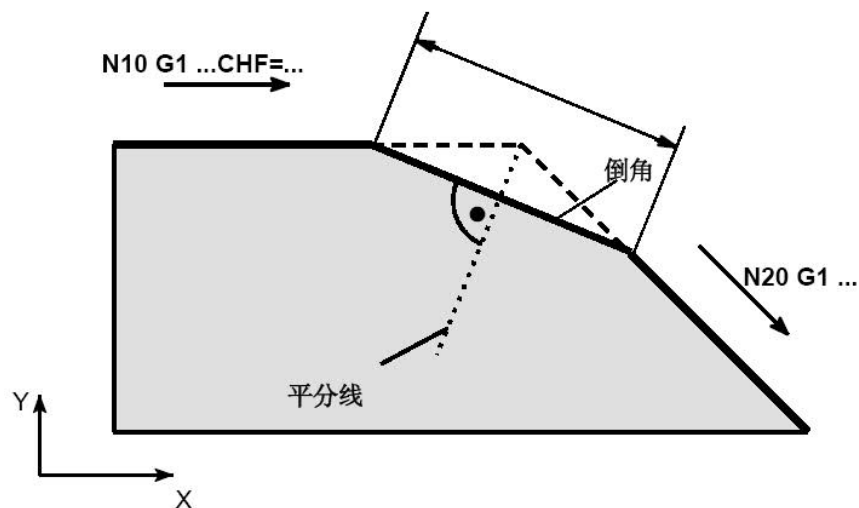
N40 M30 ; 程序结束

6. 简化轨迹计算

在直线与圆弧轨迹之间进行交接的时候,无论是相交还是相切都涉及到大量而繁琐的计算,使用下文所述的功能指令,则可以极大地简化手工编程过程中人工计算的工作量。

6.1 自动倒角

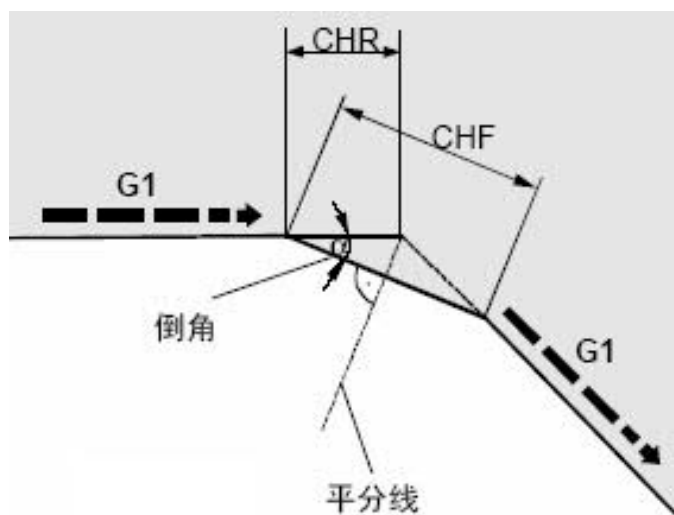
拐角处自动倒角功能: CHF、CHR



例: N10 G1 X..... CHF=5 ; 倒角三角形底边长度为 5 毫米

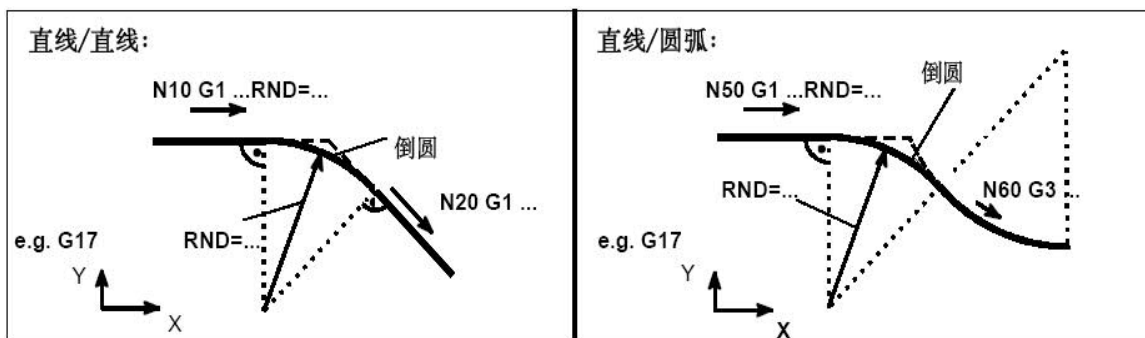
N20 G1 X..... Y.....

(若将 CHF=5 改为 CHR=5, 则倒角三角形斜边长度为 5 毫米)



6.2 自动倒圆

拐角处自动倒圆功能 RND, 可以在任意两个插补轨迹之间进行给定半径的圆弧过渡, 且过渡圆弧自动与前后两个轨迹都保持相切。



N10 G1 X... RND=8; 倒圆半径 8 毫米

N20 G1 X... Y...

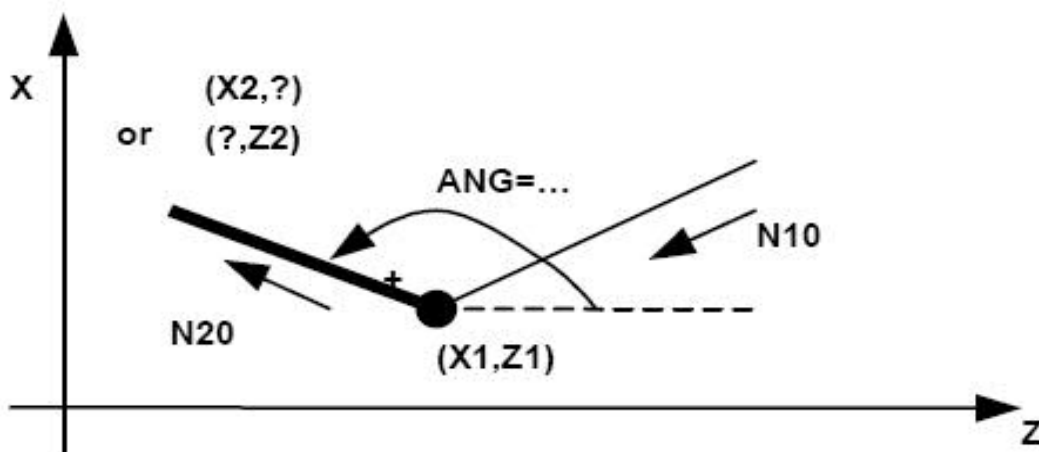
... ..

N50 G1 X... RND=7.3; 倒圆半径 7.3 毫米

N60 G3 X... Y..... CR=...

6.3 角度编程

角度编程指令 **ANG** 可以在基本平面内某一个坐标轴缺失的情况下，用该直线段与水平方向的夹角来代替。



N10 G1 X1 Z1

N20 X2 ANG=... ; Z2 坐标未知，但是该线段的角度已知

或者：

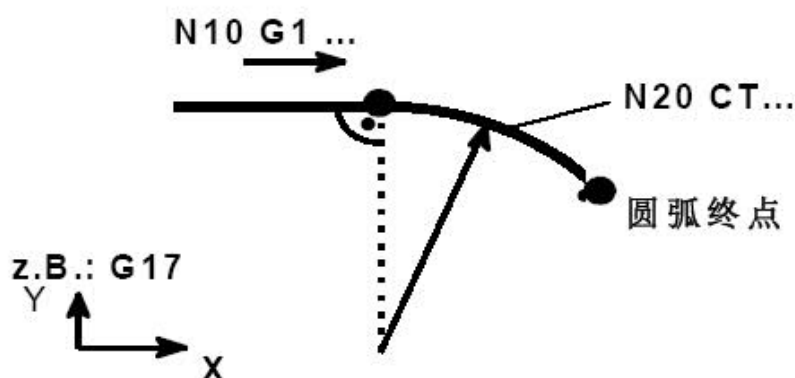
N10 G1 X1 Z1

N20 Z2 ANG=... ; Z1 坐标未知，但是该线段的角度已知

6.4 平面相切圆弧

如果要在平面中做出一条与前一条插补轨迹（G01、G02/G03）在终点处以相切的方式进行衔接的圆弧轨迹是相当麻烦的。因为这个圆弧只知道圆弧的起点和终点，圆弧的方向（顺时针/逆时

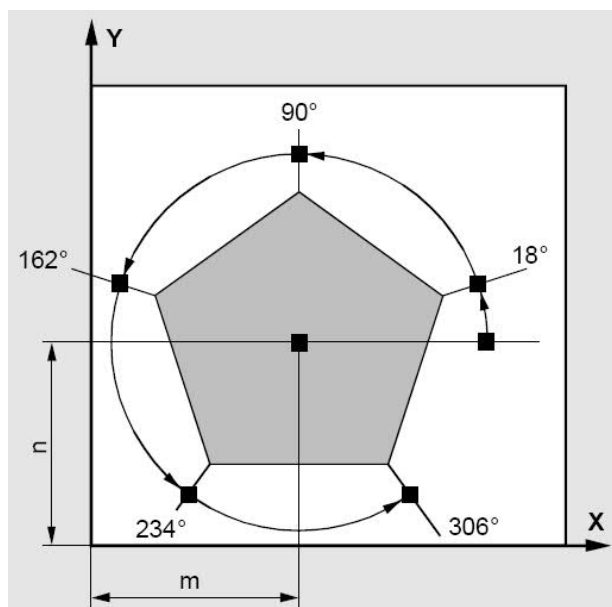
针)未知,圆弧的半径未知,圆心的位置也不知道,而这一切都需要根据“圆弧与前一轨迹在起点处相切”这个条件自行计算。



然而,使用 CT 圆弧指令这个指令,上述这些问题便会迎刃而解。只需通过“CT+圆弧终点坐标”的指令格式,就可以自动生成一条:以前一轨迹的终点为圆弧起点,并且与该轨迹相切的圆弧。圆心和半径由系统自动运算。

7. 极坐标运动指令

对于围绕某一个中心点,并且按照一定的半径尺寸进行分布的轮廓,用极坐标描述坐标点的位置比直角坐标更为直接,更加方便。

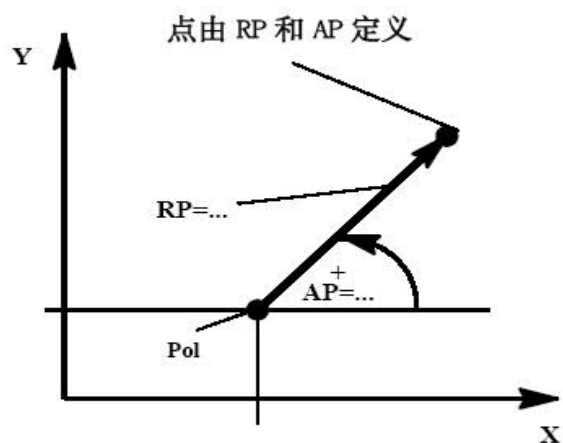


定义极点:

G110 以刀具当前点位置为基准

G111 以当前工件原点为基准

G112 以当前极点为基准



极坐标参数:

AP: 极角度

RP: 极半径

极坐标的极点一旦定义完成, 就可以与直角坐标同时使用, 不必在直角坐标系与极坐标系之间来回切换。

例如:

... ..

G111 X35 Y35; 根据当前直角坐标系的原点定义极坐标系的极点

G00 X0 Y0 Z50; 通过直角坐标系进行刀具快速定位

MCALL CYCLE82 (2, 0, 1, -13.5, ,0); 模态调用钻削循环

G00 AP=18 RP=30; 用极坐标指定钻削的位置

AP=90

AP=162

AP=234

AP=306

MCALL; 钻削循环的复位。

G00 Z100

... ..

8.作者信息

教程作者: 李晓晖