

常见数控系统宏程序编程方法、技巧与实例

FANUC

数控系统宏程序

编程方法、技巧与实例

张运强 穆瑞 编著



含1CD



机械工业出版社
CHINA MACHINE PRESS

常见数控系统宏程序编程方法、技巧与实例

FANUC 数控系统宏程序 编程方法、技巧与实例

张运强 穆 瑞 编著



机械工业出版社

本书以应用为主,以 FANUC 0i 数控系统为蓝本,详细介绍宏程序基础知识,通过大量实例分别讲述数控车床、数控铣床宏程序编制方法及技巧,使读者从简单的宏程序入门逐步了解、熟悉并掌握宏程序,将它应用到生产加工中去。内容包括数控技术基础、宏程序概述、宏程序理论基础、用户宏功能、数控车床宏程序应用实例、数控铣床宏程序应用实例。本书宏程序需要主程序调用时赋初始值,以便进行产品系列化加工。随书赠送含第 5、6 章的多媒体动画文件的光盘。

本书是一本实用性很强的数控技术用书,可为企业数控机床操作工和程序员、数控技术专业师生提供宏程序解决具有非圆曲线、列表曲线及曲面类零件的编程方案。

图书在版编目(CIP)数据

FANUC 数控系统宏程序编程方法、技巧与实例 / 张运强, 穆瑞编著.
—北京: 机械工业出版社, 2011.6 (2013.5 重印)
(常见数控系统宏程序编程方法、技巧与实例)
ISBN 978-7-111-34860-3

I. ①F… II. ①张… ②穆… III. ①数控机床—程序设计
IV. ①TG659

中国版本图书馆 CIP 数据核字 (2011) 第 100756 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 周国萍 责任编辑: 周国萍 曾 红

版式设计: 张世琴 责任校对: 常天培

封面设计: 路恩中 责任印制: 杨 曦

北京圣夫亚美印刷有限公司印刷

2013 年 5 月第 1 版第 2 次印刷

169mm×239mm·21 印张·407 千字

4001—4800 册

标准书号: ISBN 978-7-111-34860-3

ISBN 978-7-89451-982-5 (光盘)

定价: 48.00 元 (含 1CD)

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

策划编辑 (010) 88379733

电话服务

网络服务

社服 务 中 心: (010) 88361066 教 材 网: <http://www.cmpedu.com>

销 售 一 部: (010) 68326294 机 工 官 网: <http://www.cmpbook.com>

销 售 二 部: (010) 88379649 机 工 官 博: <http://weibo.com/cmp1952>

读者购书热线: (010) 88379203 封面无防伪标均为盗版

前 言

随着数控机床的普及，越来越多的企业采用数控设备进行零件的加工，对于形状简单的零件，计算比较简单，采用普通的编程方法就可以完成任务。但对于具有非圆曲线、列表曲线及曲面类的零件，用普通的方法进行程序的编制就存在一定的难度，虽然采用 CAD/CAM 软件能够自动生成加工程序，但程序段过长，且加工的效率低，而采用宏程序编程就可以有效解决上述问题，且编制出来的程序反应迅速，加工效率很高。

另外每两年一次的全国数控技能大赛，吸引了高校师生和企业职工的参与。宏程序是其中考查的一点，难度较大。

对读者来说，通过参考本书的宏程序，编写宏程序将不再头痛。特别是针对与本书实例图形相同而尺寸不同的零件时，只需要将具体变量初始值编写到主程序调用宏程序即可，对使用其他数控系统的读者也有一定的参考价值。

本书由连云港职业技术学院张运强、穆瑞老师编著，其中穆瑞老师负责编写第 1 至第 5 章，张运强老师负责第 6 章的编写以及全书的统稿。

由于编著者水平有限，时间仓促，书中难免有错误和不足之处，敬请广大读者批评指正。

编著者

目 录

前言

第 1 章 数控技术基础.....1

1.1 数控技术与数控机床的概念.....1

1.2 数控机床的发展.....1

1.3 数控机床工作原理.....2

1.3.1 数控机床的组成及其外形.....2

1.3.2 数控系统的主要功能.....3

1.3.3 数控机床的工作原理.....4

1.4 数控机床编程基础知识.....5

1.5 数控机床坐标系.....8

第 2 章 宏程序概述.....11

2.1 数控编程技术的应用现状.....11

2.2 宏程序编程的技术特点.....12

2.3 宏程序与普通程序的对比.....13

2.4 宏程序与 CAD/CAM 软件生成程序的加工性能对比.....14

2.4.1 宏程序编程的特点.....14

2.4.2 影响自动编程加工精度的因素.....15

2.4.3 自动编程与宏程序加工速度的区别.....15

2.5 学习好宏程序编程的意义.....15

第 3 章 宏程序理论基础.....17

3.1 FANUC 0i 系统的用户宏程序.....17

3.2 变量.....17

3.2.1 变量的表示.....17

3.2.2 变量的类型.....17

3.2.3 变量值的范围.....18

3.2.4 小数点的省略.....18

3.2.5 变量的引用.....18

3.2.6 未定义的变量.....19

3.3 系统变量.....20

3.3.1 接口（输入/输出）信号.....20

3.3.2 刀具补偿值.....21

3.3.3 宏程序报警.....22

3.3.4 停止和信息显示.....22

3.3.5 时间信息.....23

3.3.6 自动运行控制.....23

3.3.7 已加工的零件数.....24

3.3.8 模态信息.....24

3.3.9 当前位置信息.....25

3.3.10 工件坐标系补偿值（工件零点偏移值）.....26

3.4 算术和逻辑运算.....27

3.5 赋值与变量.....32

3.6 转移和循环.....32

3.6.1 无条件转移（GOTO 语句）.....33

3.6.2 条件转移（IF 语句）.....33

3.6.3 循环（WHILE 语句）.....34

第 4 章 用户宏功能.....37

4.1 用户宏程序调用指令 A.....37

4.1.1 宏程序模态调用与取消（G66、G67）.....37

4.1.2 子程序调用（M98）.....37

4.1.3 用 M 代码调用子程序.....37

4.1.4 用 T 代码调用子程序.....38

4.2 用户宏程序本体.....39

4.2.1 用户宏程序本体的结构.....39

4.2.2 变量的表示和引用.....39

4.2.3 变量的种类	40	5.3 圆锥曲线加工实例	60
4.2.4 宏程序的运算和控制指令	40	5.3.1 双曲线轮廓加工	60
4.3 用户宏程序调用指令 B	41	5.3.2 抛物线轮廓加工	65
4.3.1 宏程序非模态调用 (G65)	42	5.3.3 椭圆轮廓加工	69
4.3.2 宏程序模态调用与取消 (G66、G67)	45	5.4 螺纹加工实例	74
4.3.3 用 G 代码调用宏程序 (G <g>)	46	5.4.1 梯形螺纹的基本知识	74
4.3.4 用 M 代码调用宏程序 (M <m>)	47	5.4.2 单线梯形螺纹加工	84
4.3.5 用 M 代码调用子程序	48	5.4.3 多线梯形螺纹加工	90
4.3.6 用 T 代码调用子程序	49	5.4.4 变螺距螺纹加工	93
4.4 宏程序语句和 NC 语句	50	第 6 章 数控铣床宏程序应用	
4.4.1 宏程序语句和 NC 语句的 定义	50	实例	98
4.4.2 宏程序语句和 NC 语句的 异同	50	6.1 规则形状加工	98
4.4.3 宏程序语句的处理	50	6.1.1 实例一圆柱体加工	98
4.4.4 用户宏程序的存储	50	6.1.2 实例二内外锥体	107
4.5 用户宏程序的使用限制	51	6.1.3 实例三棱柱加工	125
4.6 外部输出指令	52	6.1.4 实例四棱锥加工	132
4.6.1 打开指令 POPEN	52	6.1.5 实例五球体加工	145
4.6.2 数据输出指令 BPRNT	52	6.1.6 实例六椭圆球面加工	156
4.6.3 数据输出指令 DPRNT	53	6.1.7 实例七圆环面加工	166
4.6.4 关闭指令 PCLOS	54	6.1.8 实例八铣螺纹	172
4.6.5 要求的设定	54	6.2 非圆曲线轮廓零件加工	174
4.7 中断型用户宏程序	54	6.2.1 实例一椭圆	175
4.7.1 指令格式	54	6.2.2 实例二双曲线	177
4.7.2 指定方法说明	55	6.2.3 实例三抛物线	179
4.7.3 从用户宏程序中断返回	55	6.2.4 实例四正弦曲线	181
第 5 章 数控车床宏程序应用		6.2.5 实例五余弦曲线	184
实例	57	6.2.6 实例六正切曲线	186
5.1 概述	57	6.2.7 实例七等速螺旋线	188
5.2 数控编程中的数学处理	57	6.3 孔系加工	190
		6.3.1 实例一圆周钻孔	190
		6.3.2 实例二定角度均布孔	193
		6.3.3 实例三矩阵孔加工	194
		6.3.4 实例四三角均布孔加工	199
		6.3.5 实例五可变深孔加工	202
		6.4 矩形加工	207

6.4.1	实例一矩形上平面加工	207	6.5.7	实例七椭圆柱倒圆.....	305
6.4.2	实例二矩形槽加工	211	附录		323
6.4.3	实例三矩形倒圆加工	225	附录 A	FANUC 0i 系统常用准备功	
6.4.4	实例四矩形倒角加工	240		能代码	323
6.5	倒角倒圆加工	253	附录 B	FANUC 0i -TC 系统常用辅	
6.5.1	实例一内外圆柱倒角	253		助功能代码.....	324
6.5.2	实例二内外圆柱倒圆	261	附录 C	FANUC 0i -MC 系统常用准	
6.5.3	实例三内外形倒角	270		备功能代码.....	325
6.5.4	实例四内外形倒圆	274	附录 D	可变更加工坐标系指令 G10	
6.5.5	实例五斜面加工	280		简介	327
6.5.6	实例六圆柱面加工	287	参考文献		328

第 1 章 数控技术基础

1.1 数控技术与数控机床的概念

1. 数控技术

数控技术是数字控制（Numerical Control, NC）技术的简称，是指用数字化信号对设备运行及其加工过程进行控制的一种自动化技术，它是一种可编程的自动控制方式。随着科学技术的发展，数控系统也采用专用或通用计算机及控制软件与相关的电气元部件一起来实现数控功能，称为计算机数控（CNC）系统。

2. 数控机床

装备了数控系统的机床称为数控机床。数控机床在计算机控制系统的控制下按照一定的加工指令控制主轴系统、进给系统和刀具库系统、冷却系统等设备的工作。

1.2 数控机床的发展

自 1952 年第一台数控铣床在美国诞生以来，随着电子技术、计算机技术、自动控制和精密测量技术的发展，数控机床得到迅速的发展和更新换代。

数控机床的发展先后经历了电子管（1952 年）、晶体管（1959 年）、小规模集成电路（1965 年）、大规模集成电路及小型计算机（1970 年）和微型计算机（1974 年）等五代数控系统。前三代系统是采用电子线路实现的硬件式数控系统，一般称为普通数控系统，简称 NC。第四代和第五代系统是采用微处理器及大规模或超大规模集成电路组成的软件式数控系统，称为现代数控系统，简称 CNC（第四代）和 MNC（第五代）。由于现代数控系统的控制功能大部分由软件技术来实现，因而硬件得到进一步简化，系统可靠性提高，功能更加灵活和完善。目前现代数控系统几乎完全取代了以往的普通数控系统。

随着数控系统的不断更新换代，数控机床的品种也得以不断发展，产量也不断地提高。目前，世界数控机床的品种已超过 1500 种，几乎所有品种的机床都实现了数控化。

我国数控机床的研制始于 1958 年，由清华大学研制出了最早的样机。1966 年我国诞生了第一台用直线—圆弧插补的晶体管数控系统。1970 年初研制成功集

成电路数控系统。1980 年以来通过研究和技术引进,我国数控机床发展很快,现已掌握了 5~6 轴联动、螺距误差补偿、图形显示和高精度伺服系统等多项关键技术。

1.3 数控机床工作原理

1.3.1 数控机床的组成及其外形

传统观点认为数控机床由程序载体,输入、输出设备,数控系统,伺服系统,机床本体等组成。但现代数控机床的数控系统都采用模块化结构,伺服系统中的伺服单元和驱动装置为数控系统中的一个子系统,输入、输出装置也为数控系统中的一个模块,所以现在的观点认为数控机床主要由计算机数控系统和机床本体组成。

(1) 输入装置

数控机床是按照编程人员编制的工件加工程序运行的。在工件加工程序中,包括机床上刀具和工件的相对运动轨迹、工艺参数(进给量、主轴转速等)和辅助运动等。通常编程人员将工件加工程序以一定的格式和代码存储在一种载体上,如录音磁带、软磁盘或硬盘等。通过数控机床的输入装置,将程序信息输入到数控装置内。

输入装置的作用是将程序载体内有关加工的信息读入数控装置。根据程序载体的不同,输入装置可以是录音机或软盘驱动器等。

数控机床还可以不用任何载体,通过数控机床操作面板上的键盘,用手工将工件加工程序输入数控装置;或者将存储在计算机硬盘上的工件加工程序传送到数控装置。

(2) 计算机数控装置

数控装置是数控机床的核心。它根据输入的数据,完成数值计算、逻辑判断、输入输出控制等。数控装置一般由专用(或通用)计算机、输入输出接口板及可编程序控制器(PLC)等组成。可编程序控制器主要用于对数控机床辅助功能、主轴转速功能和刀具功能的控制。

(3) 伺服系统

伺服系统包括主轴伺服单元、进给伺服单元、机床控制线路、功率放大线路及驱动装置,它接受数控装置发来的各种动作命令,驱动数控机床传动系统的运动。它的伺服精度和动态响应是影响数控机床的加工精度、表面质量和生产率的重要因素之一。

(4) 测量装置

测量装置的作用是通过位置传感器将伺服电动机的角位移或数控机床执行机构的直线位移转换成电信号，输送给数控装置，使之与指令信号进行比较，并由数控装置发出指令，纠正所产生的误差，使数控机床按工件加工程序要求的进给位置和速度完成加工。

（5）机床本体

机床本体包括：主传动系统、进给系统以及辅助装置等。对于加工中心，还有存放刀具的刀库、自动换刀装置（ATC）和自动托盘交换装置等。与传统的机床相比，数控机床的结构强度、刚度和抗振性，传动系统和刀具系统的部件结构，操作机构等方面都发生了很大的变化，其目的是为了满足不同数控技术的要求和充分发挥数控机床的效能。

1.3.2 数控系统的主要功能

（1）多轴控制功能

控制系统可以控制的坐标轴数目指的是数控系统最多可以控制多少个坐标轴，其中包括平动轴和回转轴。基本平动坐标轴是 X、Y、Z 轴，基本回转坐标轴是 A、B、C 轴。联动轴是指数控系统按照加工的要求可以控制同时运动的坐标轴的数目。如某型号的数控机床具有 X、Y、Z 三个坐标轴运动方向，而数控系统只能同时控制两个坐标（XY、YZ、XZ）方向的运动，则该机床的控制轴数为三轴（称为三轴控制），而联动轴数为两轴（称为两轴联动）。

（2）插补功能

指数控机床能够实现的运动轨迹，如直线、圆弧、螺旋线、抛物线、正弦曲线等。数控机床的插补功能越强，说明加工的轮廓种类越多。

（3）进给功能

包括快速进给（空行程移动）、切削进给、手动连续进给、点动、进给量调整（倍率开关）、自动加减速功能等性能。进给功能与伺服驱动系统的性能有很大的关系。

（4）主轴功能

可实现恒转速、恒线速度、定向停车及转速调整（倍率开关）等功能。恒线速度指主轴可以自动变速，使得刀具对工件切削点的线速度保持不变。主轴定向停车功能主要用于数控机床在换刀、精镗等工序退刀前，对主轴进行精确定位，以便于退刀。

（5）刀具功能

指在数控机床上可以实现刀具的自动选择和自动换刀。

（6）刀具补偿功能

包括刀具位置补偿、半径补偿和长度补偿功能。半径补偿中有车刀的刀尖半

径、铣刀半径的补偿，长度补偿中有车刀长度变化的补偿。

(7) 机械误差补偿功能

指系统可以自动补偿机械传动部件因间隙产生的误差的功能。

(8) 操作功能

数控机床通常有单程序段运行、跳段执行、连续运行、试运行、图形模拟仿真、机械锁住、暂停和急停等功能，有的还有软件操作功能。

(9) 程序管理功能

指对加工程序的检索、编制、插入、删除、更名、锁住、在线编辑（即后台编辑，在执行自动加工的同时进行编辑）以及程序的存储通信等。

(10) 图形显示功能

在显示器（CRT）上进行二维或三维、单色或彩色的图形显示。图形可以进行缩放、旋转，还可以进行刀具轨迹动态显示。

(11) 辅助编辑功能

如固定循环、镜像、图形缩放、子程序、宏程序、坐标轴旋转、极坐标等功能，可减少手工编程的工作量和难度。

(12) 自诊断报警功能

指数控系统对其软件、硬件故障的自我诊断能力。这项功能可以用于监视整个机床和整个加工过程是否正常，并在发生异常时及时报警。

(13) 通信功能

现代数控系统中一般都配有 RS232 接口或 DNC 接口，可以与计算机进行信号的高速传输。高档数控系统还可与 MAP 或 INTERNET 相连，以适应 FMS、CIMS 的要求。

1.3.3 数控机床的工作原理

如图 1-1 所示，首先根据零件图样进行工艺分析，确定工艺方案，依据数控系统的规定指令，编制零件的加工程序。视零件结构的复杂程度，可以采用手工或计算机编程，程序较小时，可以直接在机床操作面板的输入区域操作；程序较大时，也可以在装有程序软件的普通计算机上进行。编程软件国内一般采用模拟软件和专用软件，经过相应的后处理，生成加工程序。再通过机床控制系统上的接口或其他存储介质（软盘、光盘等），把生成的加工程序输入到机床的控制系统中。进入数控装置的信息，经过一系列处理和运算转变成脉冲信号。有的信号输送到机床的伺服系统，通过伺服机构处理，传到驱动装置（主轴电动机，步进或交、直流伺服电动机），使刀具和工件严格执行零件加工程序规定的运动；有的信号送到可编程序控制器，可以控制机床的其他辅助运动，如主轴和进给运动的变速、液压或气动装夹。

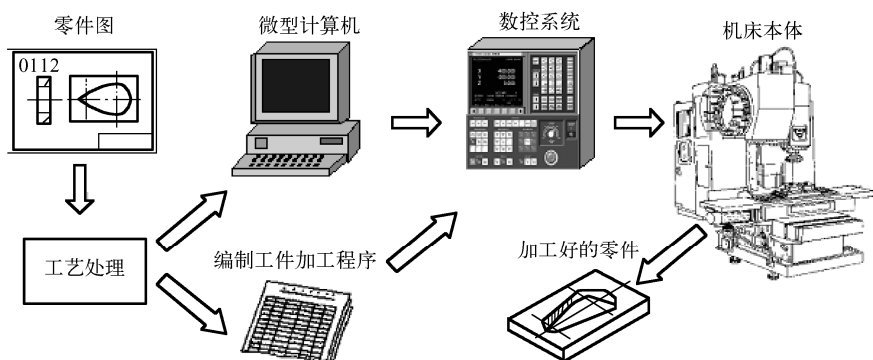


图 1-1 数控机床的工作过程

1.4 数控机床编程基础知识

数控编程是实现零件数控加工的关键环节，它包括从零件分析到获得数控加工程序的全过程。

1. 数控编程的内容

一般来说，数控编程包括以下工作：

(1) 分析零件图，制定加工工艺方案

根据零件图样，对零件的形状、尺寸、材料、精度和热处理要求等进行工艺分析，合理选择加工方案，确定工件的加工工艺路线、工序及切削用量等工艺参数，确定所用机床、刀具和夹具。

(2) 数学处理

根据零件的几何尺寸、工艺要求及编程的方便，设定坐标系，计算工件粗、精加工的轮廓轨迹，获得刀位数据。数控系统一般具有直线和圆弧插补功能，所以对于由直线和圆弧组成的形状简单的零件轮廓加工，只需计算出几何元素的起点、终点、圆弧的圆心、两几何元素的交点或切点坐标值即可，有些要计算刀具中心的运动轨迹；对于由非圆曲线或曲面组成的形状复杂的零件，需要用直线段或圆弧段来逼近曲线，根据加工精度的要求，计算出节点坐标，这个工作一般使用计算机完成。

(3) 编写零件加工程序

根据制定的加工工艺路线、切削用量、刀具补偿量、辅助动作及刀具运动轨迹等条件，按照机床数控系统规定的功能指令代码及程序格式，逐段编写加工程序。

(4) 记录程序并输入到数控机床

记录编制好的程序，并传输到数控机床中，这个工作可通过手工在操作面板

直接输入，或利用通信方式输入，由传输软件把计算机上的加工程序传输到数控机床。

（5）程序校验和试切

输入到数控系统的加工程序在正式加工前需要进行验证，以确保程序正确。通常可采用机床空运行的方法，检查机床动作和运动轨迹是否正确；在有图形功能的数控机床上，可以利用模拟加工的图形显示来检查运行轨迹的正确性。需要注意的是这些方法只能检验运动轨迹是否正确，不能检验被加工零件的精度。因此，需要进行零件的首件试切，当发现加工的零件不符合加工技术要求时，分析产生加工误差的原因，找出问题，修改程序或采取尺寸补偿等措施。

2. 数控编程方法

（1）手工编程

手工编程就是指数控编程内容的工作全部由人工完成。对形状比较简单的工件，其计算量小，程序短，手工编程快捷、简便。对形状复杂的工件采用手工编程有一定难度，有时甚至无法实现。一般说来，由直线和圆弧组成的工件轮廓采用手工编程，非圆曲线、列表曲线组成的轮廓采用自动编程。

（2）自动编程

自动编程就是利用计算机专用软件完成数控机床程序编制工作。编程人员只需根据零件图样的要求，使用数控语言由计算机进行数值计算和工艺参数处理，自动生成加工程序，再通过通信方式传入数控机床。

3. 程序格式

（1）字符与代码

字符是用于组织、控制或表示数据的一些符号，进行信息交换，数字、字母、标点符号、数学运算符都可以用作字符，常规加工程序应用四种字符：英文字母、数字和小数点、正负号、功能字符。

（2）程序字（简称字或指令字）

字是一套可以作为一个信息单元进行存储、传递和操作的有规定次序的字符，字符的个数即为字长。常规加工程序中的字都是由英文字及随后的数字组成，这个英文字称为地址符，地址符与后续数字之间可有正负号，如 X30 Z-25。

常用地址符功能见表 1-1。

（3）字的几种功能

1) 语句号 N（也称为程序段号）。程序是一句一句编写的，一句程序称为一个程序段。程序段号字用以识别每一程序段，由地址符 N 和若干位数字组成。例如，N40 表示该程序段的语句号为 40。

2) 准备功能字 G（又称 G 功能、G 指令、G 代码）。准备功能是用来建立机床或数控系统工作方式的一种命令，使数控机床做好某种操作准备，用地址符 G

和两位或三位数字表示。需要指出的是，不同生产厂家数控系统 G 指令的功能相差大，编程时必须遵照机床使用说明书进行。

表 1-1 常用地址符功能

功能	地址符	意义
程序号	O, P	程序号，子程序号的指令
顺序段号	N	程序段号
准备功能	G	指令动作方式
尺寸字	X, Y, Z	坐标轴的移动指令
	A, B, C, U, V, W	附加坐标轴的移动指令
	I, J, K	圆弧中心坐标
	R	圆弧半径
进给功能	F	进给速度指令
主轴功能	S	主轴转速指令
刀具功能	T	刀具编号
辅助功能	M, B	主轴、切削液的开/关，工作台分度等

3) G 指令分模态指令（续效指令）和非模态指令（非续效指令）。非模态指令只在本程序段中有效，模态指令可在连续几个程序段中有效，直到被相同组别的指令取代。指令表中标有相同数字或字母的为一组，如 G00、G01、G02、G03、G04，其中 G04 为非模态指令，其余为模态指令。

4) 尺寸字。由地址符、符号（+、-）、绝对（或相对）数值组成。尺寸字的地址符有 X、Y、Z、U、V、W、P、Q、R、A、B、C、I、J、K、D、H 等。例如，X15 Y-20，其中“+”可省略。

5) 进给功能字 F。表示加工时的进给速度，由地址符 F 和后面的若干位数字组成。

6) 主轴功能字 S。表示数控机床主轴转速，由地址符 S 和后面的若干位数字组成。

7) 刀具功能字 T。由地址符 T 和后面的若干位数字组成。数字表示刀号，数字位数由数控系统决定。

8) 辅助功能字 M（又称 M 功能、M 指令、M 代码）。用来控制机床辅助动作或系统的开关功能，由地址符和后面的两位数字组成。

(4) 程序段格式

零件的加工程序由若干个程序段组成。程序段格式是指一个程序段中字、字符、数据的书写规则，目前使用最多的是“字—地址”程序段格式，如图 1-2 所示。

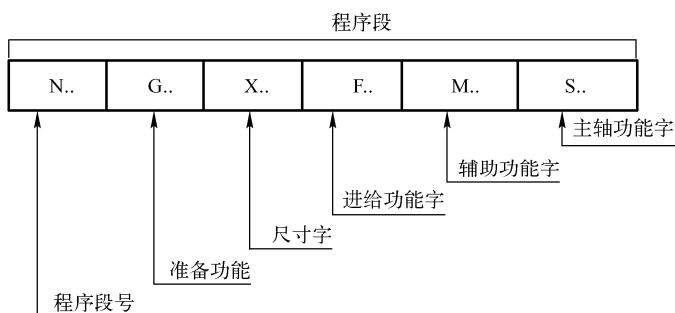


图 1-2 程序段格式

字—地址程序段由程序段号字、数据字和程序段结束符组成。各字后有地址，字的排列顺序要求不严格，数据的位数可多可少，不需要的字以及与上一程序段相同的续效字可以不写。

例：N30 G01 X50 Z-20 F100 S400 T01 M03

需要说明的是，数控加工程序内容、指令和程序段格式虽然在国际上有很多标准，实际上并不是完全统一，所以在编制加工程序前，必须详细了解机床数控系统编程说明书中的具体指令格式和编程方法。

1.5 数控机床坐标系

为简化编程和保证程序的通用性，对数控机床的坐标轴和方向命名制定了统一的标准。国际标准化组织以及一些工业发达国家都先后制定了数控机床坐标系和运动命名的标准。

1. 坐标系和运动方向命名的原则

1) 数控机床中统一规定采用右手笛卡儿坐标系命名，如图 1-3 所示。图中大拇指的指向为 X 轴的正方向，食指指向为 Y 轴的正方向，中指指向为 Z 轴的正方向。

2) 坐标系中的各个坐标轴与机床的主要导轨相平行。

3) 机床在加工过程中不论是刀具移动还是工件移动，都一律假定工件相对静止不动，而刀具在移动，并规定刀具远离工件的运动方向为坐标轴正方向。

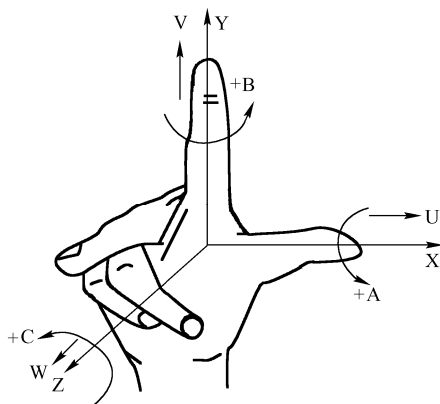


图 1-3 右手笛卡儿坐标系

4) 旋转运动。围绕 X、Y、Z 轴旋转的圆周进给坐标轴分别用 A、B、C 表示, 其正方向用右手螺旋法则确定, 以大拇指指向+X、+Y、+Z 方向, 则食指和中指等的指向是圆周进给运动的+A、+B、+C。

2. 坐标运动的规定

1) Z 坐标运动由传递切削动力的主轴所决定。与主轴轴线平行的标准坐标轴即为 Z 坐标, 如数控车床、数控立式镗铣床等。

若机床没有主轴(如数控刨床等), 则 Z 坐标垂直于工件主要装夹面; 若机床有几个主轴, 可选择一垂直于工件装夹面的主轴作为主要主轴, 并以它确定 Z 坐标, 如数控龙门铣床。

Z 坐标的正方向是增加刀具和工件之间距离的方向, 如在钻、镗加工中, 钻入或镗入工件的方向是 Z 坐标的负方向。

2) X 坐标的运动。X 坐标运动是水平的, 它平行于工件装夹面, 是刀具或工件定位平面内运动的主要坐标。

在没有回转刀具和回转工件的机床上(如牛头刨床), X 坐标平行于主要切削方向, 并且以该方向为正方向。

在有回转工件的机床上, 如车床、磨床等, X 坐标方向是在工件径向, 而且平行于横向滑座。对于安装在横向滑座的主要刀架上的刀具, 离开工件回转中心的方向是 X 坐标的正方向。

在有刀具回转的机床上(如铣床), 若 Z 坐标是水平的(主轴是卧式的), 当由主要刀具主轴向工件看时, X 运动的正方向指向右方; 如果 Z 坐标是垂直的(主轴是立式的), 当由主要刀具主轴向立柱看时, X 运动的正方向指向右方。

3) Y 坐标的运动。正向 Y 坐标的运动, 根据 X 和 Z 的运动, 按照右手笛卡儿坐标系来确定。

4) 机床坐标系的原点。机床坐标系的原点位置是任意选择的, 由生产厂家调定。A、B、C 的运动原点(0° 的位置)也是任意的, 但 A、B、C 原点的位置最好选择与 X、Y、Z 坐标平行。

5) 附加坐标。如果在 X、Y、Z 主要直线运动之外另有第二组平行于它们的坐标运动, 就称为附加坐标。它们分别被规定为 U、V 和 W。

如果在第一组回转运动 A、B、C 之外, 还有平行或不平行于 A、B、C 的第二组回转运动, 可指定为 P、Q 或 R。

3. 机床坐标系、机床原点和机床参考点

1) 机床坐标系。机床坐标系是机床上固有的坐标系, 并设有固定的坐标原点, 就是机床原点, 又称机械原点, 即 $X=0$ 、 $Y=0$ 、 $Z=0$ 的点。对某一具体机床来说, 在经过设计、制造和调整, 这个原点便被固定下来, 它是机床上固定的点。

2) 机床参考点。为了正确地建立机床坐标系, 通常在每个坐标轴的移动范围

内设置一个机床参考点，又称机械原点，即 $X=0$ 、 $Y=0$ 、 $Z=0$ 的点，其固定位置由各轴向的机械挡块来确定。一般机床开机后，通常要进行机动或手动回参考点以建立机床坐标系。

机床参考点可以与机床零点重合也可以不重合，通过参数指定机床参考点到机床零点的距离。机床回到了参考点位置也就知道了该坐标轴的零点位置，找到所有坐标轴的参考点，机床坐标系就建立起来了。机床参考点在数控机床出厂时就已经调好，并记录在说明书中供用户编程使用。一般情况下不允许随意变动。

3) 工件坐标系与原点。工件坐标系是编程人员在编程时使用的，编程人员选择工件上的某一已知点为原点（也称工件原点、程序原点），建立一个新的坐标系，称为工件坐标系。工件坐标系一旦建立便一直有效，直到被新的工件坐标系所取代。

工件坐标系的原点是人为设定的，设定的依据是要尽量满足编程简单、尺寸换算少、引起的加工误差小等条件。一般情况下，程序原点应选在尺寸标注的基准或定位基准上。对称零件或以同心圆为主的零件，编程原点应选在对称中心线或圆心上。 Z 轴的程序原点通常选在工件的表面。

第 2 章 宏程序概述

2.1 数控编程技术的应用现状

数控编程是从零件图样到获得数控加工程序的全过程。它的主要任务是计算加工走刀中的刀位点。刀位点一般取为刀具轴线与刀具表面的交点，多轴加工中还要给出刀轴矢量。

数控技术是 20 世纪 40 年代中后期为适应加工复杂外形零件而发展起来的一种自动化控制技术。数控加工以加工形状复杂、改型频繁、小批量生产的零件为主。其主要优点是：加工精度高，生产效率高，自动化程度高，使用数字信息，便于计算机控制。尤其是近年来随着 CAD/CAM 技术的发展及应用，数控加工技术的优越性更加明显。数控编程是目前 CAD/CAPP/CAM 系统中最能明显发挥效益的环节之一，尤其在实现设计加工自动化、提高加工精度和加工质量、缩短产品研制周期等方面发挥着重要作用，在航空、机械、电子等领域有着大量的应用。由于生产实际的强烈需求，国内外都对数控编程技术进行了广泛的研究，并取得了丰硕的成果。为了解决数控加工中的程序编制问题，20 世纪 50 年代，MIT 设计了一种专门用于机械零件数控加工程序编制的语言，称为 APT。采用 APT 语言编制数控程序具有程序简练、走刀控制灵活等优点，使数控加工编程从面向机床指令的“汇编语言”级，上升到面向几何元素。但 APT 仍有许多不便之处：采用语言定义零件几何形状时难以描述复杂的几何形状，缺乏几何形状直观性；缺少对零件形状、刀具运动轨迹的直观图形显示和刀具轨迹的验证手段；难以和 CAD 数据库和 CAPP 系统有效连接；不容易做到高度的自动化和集成化。针对 APT 语言的缺点，1978 年，法国达索飞机公司开始开发集三维设计、分析 NC 加工一体化的系统，称为 CATIA。随后很快出现了如 EUCLID、UGII、INTERGRAPH、Pro/Engineering、Master CAM 及 NPU/GNCP 等系统，这些系统都有效地解决了几何造型、零件几何形状的显示，交互设计、修改及刀具轨迹生成，走刀过程的仿真、验证等问题，推动了 CAD 和 CAM 向一体化方向发展。到了 20 世纪 80 年代，在 CAD/CAM 一体化概念的基础上，逐步形成了计算机集成制造系统 CIMS 及并行工程 CE 的概念。虽然这些 CAD/CAM 软件都能很好地应用编程，但在反应能力方面不是很迅速，并且要求的条件也很高。目前，为了 CIMS 及 CE 发展的需要，数控编程系统正向集成化和智能化方向发展。

我国目前制造业飞速发展，与世界制造先进水平的差距在不断缩小，而数控加工技术作为现代制造技术的灵魂及核心也得到了广泛的应用，各类 CAD/CAM 软件的应用日趋普及，特别是在数控三维曲面加工中手工编程几乎已无用武之地，而强大的思维定式和使用习惯，使得编程人员不论程序大小、加工难易都习惯使用 CAD/CAM 软件来编程，手工编程似乎被遗忘了。而在学习手工编程时只是简单地学习基本的编程指令，对宏程序也是如此，原因是大家对宏程序不熟悉，往往以为宏程序深不可测而已。在实际工作中宏程序确实有广泛的应用空间，并且能够方便编程，任何数控加工只要能够用宏程序完整地表达，即使再复杂，其编程篇幅都比较精练，数控机床在执行宏程序时比 CAD/CAM 软件生成的程序更快捷，反应更迅速，使得加工效率大大提高。

2.2 宏程序编程的技术特点

数控编程方法有手工编程和自动编程两种。手工编程是指从零件图样分析、工艺处理、数据计算、编写程序单、输入程序到程序校验等步骤主要由人工完成的编程过程。它适用于点位加工或几何形状不太复杂的零件加工，以及计算较简单，程序段不多，编程易于实现的场合等。但对于几何形状复杂的零件（尤其是空间曲面组成的零件），以及几何元素不复杂但需编制程序量很大的零件，由于编程时计算数值的工作相当繁琐，工作量大，容易出错，程序校验也较困难，用手工编程难以完成，因此要采用自动编程。所谓自动编程即程序编制工作的大部分或全部由计算机完成，可以有效解决复杂零件的加工问题，也是数控编程未来的发展趋势。

尽管使用各种 CAD/CAM 软件来编程已成为数控加工潮流，但手工编程毕竟还是基础，各种疑难杂症的解决往往还要手工编程，手工编程可以使用变量编程，即参数的运用。在机械零件加工中经常遇到一些零件上有许多相同或相似的几何形状，或者形状相似的零件，在数控编程中如果把这些几何形状一一编写出来，不但程序很大，数据较多，而且出现错误也不易检查出来。为解决这一问题，总结出一种参数化数控编程方法。在编写加工程序时，把一组命令构成的某种特定功能，像子程序那样记录在存储器中，其功能可用一个命令来代表，并使用该命令调用它。其一组命令称为参数指令体。代表的命令称为参数指令。用户不必记忆参数指令体的一组命令，只要记忆代表参数指令体的参数指令即可。用参数程序的特点是参数程序中有变量，变量之间可进行运算，用参数指令给变量赋值。把某个功能作为参数程序编程时，可将变化的值、未知值作为变量编程。这种变量式编程称之为参数化编程。其特点是将有规律的形状或尺寸用最短的程序表示出来，具有极好的易读性和易修改性，编写出的程序非常简单，逻辑严密，通用

性强,而且机床执行此类程序时比 CAD/CAM 软件生成的程序更快捷,反应更迅速。同时参数化编程具有灵活性、通用性和智能性等特点。例如对于规则曲面的编程来说,使用 CAD/CAM 软件编程工作量大,程序庞大,加工参数不易修改,只要任何一个加工参数发生变化,再智能的软件也要根据变化后的加工参数重新计算刀具轨迹。尽管软件计算刀具轨迹的速度非常快,但始终是个比较麻烦的过程。而宏程序则注重把机床功能参数与编程语言结合,灵活的参数设置也使机床具有最佳的工作性能,给予操作者极大的自由空间。

从模块化加工角度看,宏程序最具有模块化的思想和物质条件,编程人员只需把零件信息、加工参数等输入到相应模块的调用语句中即可,从而使编程人员从烦琐的大量重复性的编程工作中解脱出来。

另外,宏程序几乎包含了所有的加工信息,而且简明、直观,通过简单地存储和调用,就可以很方便地重现当时的加工状态,给周期性的生产带来极大的便利。

宏程序编程可以减少数学运算过程的计算误差,提高处理能力,理想地逼近加工的曲面、曲线,提高加工精度。随着技术的发展,自动编程将逐渐取代手工编程,但宏程序简捷的特点使之依然具有使用价值。作者个人认为,宏程序的运用应该是手工编程应用中最大的亮点。

2.3 宏程序与普通程序的对比

在一般的程序编制中程序字为常量,一个程序只能描述一个几何形状,当工件形状没有发生改变只是尺寸发生改变时,只能重新进行编程,缺乏灵活性和通用性。而宏程序则是当所要加工的零件如果形状没有发生变化只是尺寸发生了一定变化时,在程序中给要发生变化的尺寸设几个变量,再加上必要的计算公式,当尺寸发生变化时只要改变这几个变量的赋值参数就可以了。这种有变量的程序叫宏程序,它是利用对变量的赋值和表达式来进行程序的编辑。

一般意义上所讲的数控指令其实是指 ISO 代码指令,即每个代码的功能是固定的,由系统生产厂家开发,使用者只需利用每个指令代码的功能,按照规定编程即可。但有时候这些简单的指令不能满足用户加工要求,因此系统提供了用户宏程序功能,使用户可以对数控系统进行一定的功能扩展。普通程序只能使用常量进行编程,而且常量之间不可以进行计算,而宏程序则可以使用变量进行编程,并可以给变量进行赋值,变量之间可以根据程序中给出的表达式进行计算。实际上是数控系统对用户的开放,也可视为用户利用数控系统提供的工具,在数控系统的平台上进行二次开发,当然这里的开放和开发都是有条件和限制的。

用户宏程序与普通程序存在一定的区别，认识和了解这些区别，将有助于宏程序的学习理解和掌握运用，表 2-1 为用户宏程序和普通程序的对比。

表 2-1 用户宏程序和普通程序的对比

普 通 程 序	宏 程 序
只能使用常量	可以使用变量，并给变量赋值
常量之间不可以运算	变量之间可以运算
程序只能顺序执行，不能跳转	程序运行可以跳转

2.4 宏程序与 CAD/CAM 软件生成程序的加工性能对比

在 CAD/CAM 软件普遍存在的今天，手工编程的应用空间日趋减小，数控行业有一种说法很流行“宏程序已经没有什么用”，其实任何数控系统都有很多指令在一般情况下用不着，那它们是否没有用呢？这显然不对，对宏程序也是如此。因为人们对宏程序不熟悉，往往误以为宏程序深不可测，其实在实际工作中，宏程序有广阔的应用空间，并且能够方便手工编程，锻炼操作者的编程能力，帮助操作者更加深入地了解自动编程的本质。

自动编程有许多优点，但是自动编程也有其不利于加工方面的问题，在加工不规律的曲面时利用自动编程确实是很好，但是在加工有规律的曲面时就不见得了，加工有规律的工件时用宏程序编程要比用自动编程软件强得多，而且宏程序比较精练，不像自动编程那么烦琐，比较长，看起来比较麻烦。

2.4.1 宏程序编程的特点

宏程序是程序编制的高级形式，程序编制的质量与编程人员的素质息息相关，宏程序里应用了大量的编程技巧。它利用数学关系表达加工刀具的选择、走刀方式的取舍等，这些都使得宏程序的精度很高，特别是对于中等难度的零件，使用宏程序加工要比自动编程加工快得多。

另外很重要的一点，CAD/CAM 所生成的程序很多，而且非常的烦琐，有的时候自动编程的程序长度可能是宏程序长度的几十倍，甚至几百倍，加工时间也会大大增加。

加工一个半径为 25mm 半球的程序，如同时采用自动编程和手工编程，采用同样的刀具、同样的加工参数，宏程序只有 20 几行，而自动编程软件编制的程序则多达 2000 多行，并且自动编程所生成的程序几乎是由 G01、G02、G03 等简单指令组成，但是后面的数字全部是小数，采用直线逼近圆或直线逼近曲线的方法，修改很麻烦，有的时候甚至可以说是无从下手。由此可以看出宏程序是精练的程

序。宏程序是手工编程，CAD/CAM 是自动编程，手工编程是自动编程的基础，在任何时候手工编程都是必须掌握的。

2.4.2 影响自动编程加工精度的因素

自动编程产生的加工程序的精度受多方面因素的影响，首先它受 CAD/CAM 软件建模时计算精度的影响，不同软件之间 CAD 图档的转换精度是不同的，其次受自动编程软件在生成 NC 刀具轨迹时计算精度的影响，再者就是后处理环节有时也会对其有影响。

自动编程生成的程序多数采用直线逼近圆或直线逼近曲线组成，表面看上去是圆或曲线，而实际上是很多的台阶或者是 N 边形，只不过是这个台阶很小或者 N 边形的 N 很大而已。在加工圆时走的轨迹不是一个整圆，精度可想而知不如轨迹为整圆的高。

2.4.3 自动编程与宏程序加工速度的区别

因为自动编程生成的程序比较烦琐，可能加工一个简单的零件，就会有几千乃至上万行的程序，而机床内部程序的存储空间是有限的，FANUC 0i 系统的标准配置一般为 128KB 或 256KB，这上千乃至上万行的程序不止 128KB 或 256KB，这就需要 DNC 方式在线加工，此时机床与计算机之间的传输速度成了影响加工速度的关键。目前的机床大多数采用的是 RS232 口的串口通信来实现在线加工，大多数系统所支持的 RS232 口最大的传输速度为 19200bit/s，即使是在最大的传输速度下，当计算精度较高、进给速度较快时，程序的传输速度也会跟不上，出现进给运动有明显的时断时续的现象，采用其他方法也不会有太大的改观。而使用宏程序加工时由于计算速度较快，不会出现加工中时断时续的现象。

2.5 学习好宏程序编程的意义

宏程序在实际生产中具有以下意义：

1) 减少编程时间，使机床具有最佳的工作性能，最大限度地提高生产效率，降低成本。机械零件绝大多数都是批量生产，在保证质量的前提下要求最大限度地提高效率以降低成本，一个零件哪怕仅仅节约 1s，成百上千个同样零件合起来节约的时间就非常可观了。

2) 优化加工工艺。加工工艺的优化主要就是程序的优化，是一个反复调整、尝试的过程，这就要求操作者能够非常方便地调整程序中的各项加工参数，而只要其中任何一项发生变化，再智能的软件也要根据变化后的加工参数重新计算刀具轨迹，过程耗时费力烦琐。宏程序在这方面就有强大的优越性，操作者无需触

动程序本身，只需对各加工参数所对应的自变量赋值做出个别调整就可以将程序调整到最优化的状态，这也是宏程序的一个突出的优点。

3) 用途广，可以进行有规律的数学运算。机械零件的形状主要是由凸台、凹槽、圆孔、斜平面、回转面组成，很少包含不规则的复杂曲面，构成的几何因素包括点、直线、圆弧、各种二次曲线（椭圆、抛物线、双曲线）以及一些渐开线，所有这些都基于三角函数、解析几何，而数学上都可以用三角函数表达式及参数方程加以表达，因此宏程序可以发挥其最大的作用。

4) 解决生产中的一些复杂加工编程问题。机械零件还有一些特殊的应用，即使采用 CAD/CAM 软件也不一定能够轻而易举地解决，如变螺距螺纹的加工和可变深度孔钻削加工等，而宏程序就可以发挥它的优势。

5) 改变编程者工艺指导思想，提高编程的工艺制定水平。无论是数控技能鉴定等级考试还是我国每两年举行一次的数控技能大赛，都不允许使用 CAD/CAM 软件编程，而只能用手工编程。在实际工作中手工编程依然存在，尤其对宏程序的运用有明确的要求。

在数控加工技术中手工编程是基础，能应用手工编程的地方尽量不使用自动编程，而且宏程序具有灵活性、通用性和智能性的特点，用宏程序编程可以直接地体现编程者的工艺指导思想。任何时候手工编程都是必须掌握的。对于加工一些具有特别规律的外形、曲面，如椭圆、椭圆锥台、凹凸球面、孔口倒角、孔口倒圆弧角、四边倒角、倒圆角等都可以使用宏程序进行编程，大大减少了编程工作量。我国这几年为提高数控技术人员的水平，每两年举行一次数控大赛，它是不允许使用自动编程的，如果不会使用宏程序编程，用普通编程的方法是无法进行比赛的，因此我们更有必要学习好数控技术中的宏程序编程技术。

第 3 章 宏程序理论基础

3.1 FANUC 0i 系统的用户宏程序

FANUC 0i 系统提供两种用户宏程序，即用户宏程序功能 A 和用户宏程序功能 B。用户宏程序功能 A 是 FANUC 系统的标准配置功能，只要配置 FANUC 系统都具备此功能，而用户宏程序功能 B 虽然不是 FANUC 系统的标准配置功能，但是绝大部分的 FANUC 系统也都支持用户宏程序功能 B。

由于用户宏程序功能 A 的宏程序需要使用“G65Hm”格式的宏指令来表达各种数学运算和逻辑关系，极不直观，且可读性非常差，因而导致在实际工作中很少有人使用它，大多数用户可能根本就不知道它的存在。本书仅对用户宏程序功能 A 作简单的介绍，不进行深入讲述，而将用户宏程序功能 B 作为重点深入介绍宏程序的相关知识。

3.2 变量

普通加工程序直接用数值指定 G 代码和移动距离。例如：G01 和 X100.0。

使用用户宏程序时，数值可以直接指定或用变量指定，当用变量时，变量值可用程序或用 MDI 面板来进行操作设定或修改。

```
#1=#2+100;  
G01 X#11 F0.3;
```

3.2.1 变量的表示

与通用的编程语言不同，用户宏程序则不允许使用变量名，变量需用变量符号“#”和后面的变量号指定。例如：#1。

表达式可以用于指定变量号，这时表达式必须封闭在括号中。例如：#[#1+#2-12]。

3.2.2 变量的类型

变量从功能上主要可以归纳为两种，即系统变量和用户变量。

系统变量（系统占用部分）：用于系统内部运算时各种数据的存储。

用户变量：包括局部变量和公共变量，用户可以单独使用。FANUC 0i 系统的变量类型见表 3-1。

表 3-1 FANUC 0i 系统的变量类型

变量号		变量类型	功 能
#0		空变量	该变量总是空，没有值能赋予该变量
用户变量	#1~#33	局部变量	局部变量只能在宏程序中存储数据，断电时该变量被初始化为空，调用宏程序时，可以在程序中对其赋值
	#100~#199 #500~#999	公共变量	公共变量在不同的宏程序中的意义相同，断电时，#100~#199 清除（初始化为空），通电时复位到“0” #500~#999 数据，即使在断电时也不清除
	#1000 以上	系统变量	系统变量用于读与写 CNC 运行时各种数据的变化

3.2.3 变量值的范围

局部变量和公共变量可以是 0 值或以下范围中的值： $-10^{47} \sim 10^{-29}$ 或 $10^{-29} \sim 10^{47}$ ，如果计算结果超出有效范围，则触发程序错误 P/S 报警 No.111。

3.2.4 小数点的省略

当在程序中定义变量值时，整数值的小数点可以省略。例如：定义#1=123；变量#1 的实际值是 123.000。

3.2.5 变量的引用

在程序中使用变量值时，应指定后跟变量号的地址。当用表达式指定变量时，必须把表达式放在括号中。例如：G01 X[#1+#2] F#3

被引用变量的值根据地址的最小设定单位自动地舍入。

例如：当 G00 X#1；以 1/1000mm 的单位执行时，CNC 把 12.3456 赋值给变量#1，实际指令值为 G00 X12.346。

改变引用变量值的符号，要把负号（-）放在#的前面。

例如：G00 X-#1。

当引用未定义的变量时，变量及地址字都被忽略。

例如：当变量#1 的值是 0，并且变量#2 的值是空时，G00 X#1 Y#2 的执行结果为 G00 X0。

从这个例子可以看出，所谓“变量的值是 0”与“变量的值是空”是两个完全不同的概念，可以这样理解：“变量的值是 0”相当于“变量的数值等于 0”，而“变量的值是空”则意味着“该变量所对应的地址根本就不存在，不生效”。

不能用变量表示的地址符有：程序号 O，顺序号 N，任选程序段跳转号/。以

下情况不能使用变量，例如：

```
O#11;      /O#22 G00 X100.0;      N#33 Y200.0;
```

另外，使用 ISO 代码编程时，可用“#”代码表示变量；若用 EIA 代码，则应用“&”代码代替“#”代码，因为 EIA 代码中没有“#”代码。

3.2.6 未定义的变量

当变量值未定义时，这样的变量成为“空”变量。变量#0 总是空变量。它不能写，只能读。

(1) 引用

当引用一个未定义的变量时，地址本身也被忽略。

当#1=<空>时	当#1=0 时
G90 X100 Z#1; ...	G90 X100 Z#1; ...
G90 X100;	G90 X100 Z0;

(2) 运算

除了用<空>赋值以外，其余情况下<空>与 0 相同。

当#1=<空>时	当#1=0 时
#2=#1 ...	#2=#1 ...
#2=<空>	#2=0
#2=#1*5 ...	#2=#1*5 ...
#2=0	#2=0
#2=#1+#1 ...	#2=#1+#1 ...
#2=0	#2=0

(3) 条件表达式

EQ 和 NE 中的<空>不同于 0。

当#1=<空>时	当#1=0 时
#1 EQ #0 ...	#1 EQ #0 ...
成立	不成立
#1 NE #0 ...	#1 NE #0 ...
成立	不成立
#1 GE #0 ...	#1 GE #0 ...
成立	成立
#1 GT #0 ...	#1 GT #0 ...
不成立	不成立

3.3 系统变量

系统变量用于读和写 CNC 内部数据，例如：刀具偏置值和当前位置数据。无论是用户宏程序功能 A 或用户宏程序功能 B，系统变量的用法都是固定的，而且某些系统变量为只读，用户必须严格按照规定使用。

系统变量是自动控制和通用加工程序开发的基础，在这里仅就系统变量的部分内容（与编程及操作相关性较大）加以介绍（表 3-2）。

表 3-2 FANUC 0i 系统变量一览表

变 量 号	含 义
#1000～#1015, #1032	接口输入变量
#1100～#1115, #1132, #1133	接口输出变量
#10001～#10400, #11001～#11400	刀具长度补偿值
#12001～#12400, #13001～#13400	刀具半径补偿值
#2001～#2400	刀具长度与半径补偿值（偏置组数≤200 时）
#3000	报警
#3001, #3002	时钟
#3003, #3004	循环运行控制
#3005	设定数据（SETTING 值）
#3006	停止和信息显示
#3007	镜像
#3011, #3012	日期和时间
#3901, #3902	零件数
#4001～#4120, #4130	模态信息
#5001, #5104	位置信息
#5201, #5324	工件坐标系补偿值（工件零点偏移值）
#7001～#7944	扩展工件坐标系补偿值（工件零点偏移值）

下面对各系统变量进行详细说明。

3.3.1 接口（输入/输出）信号

接口信号是可编程序控制器（PMC）和用户宏程序之间交换的信号，见表 3-3。

只有使用 FANUC PMC 时才能使用表 3-3 中的变量。在运算中，系统变量 #1000～#1015 和#1032 不能用作左边的项。

表 3-3 FANUC 0i 接口信号的系统变量

(参数 No.6001#0 (MIF) =0 时)

变量号	功 能
#1000~#1015 #1032	把 16 位信号从 PMC 送到用户宏程序, 变量#1000~#1015 用于按位数读取信号; 变量#1032 用于一次读取一个 16 位信号
#1100~#1115 #1132	把 16 位信号从用户宏程序送到 PMC, 变量#1100~#1115 用于按位数写信号; 变量#1132 用于一次写一个 16 位信号
#1133	变量#1133 用于从用户宏程序一次写一个 32 位的信号送到 PMC 注意: #1133 的值为-99999999~+99999999

(参数 No.6001#0 (MIF) =1 时)

变量号	功 能
#1000~#1031	把 32 位信号从 PMC 送到用户宏程序, 变量#1000~#1031 用于按位数读取信号
#1100~#1131	把 32 位信号从用户宏程序送到 PMC, 变量#1100~#1131 用于按位数读写信号
#1032~#1035	把 32 位信号从 PMC 统一输出到用户宏程序的变量 只能在-99999999~+99999999 的范围内指定
#1132~#1135	把 32 位信号统一写入到用户宏程序的变量 只能在-99999999~+99999999 的范围内指定

3.3.2 刀具补偿值

用系统变量可以读和写刀具补偿值。通过对系统变量赋值, 可以修改刀具补偿值, 见表 3-4。

表 3-4 FANUC 0i 刀具补偿存储器 C 的系统变量

补偿号	刀具长度补偿 (H)		刀具半径补偿 (D)	
	几何补偿	磨损补偿	几何补偿	磨损补偿
1	#11001 (#2201)	#10001 (#2001)	#13001	#12001
2	#11002 (#2202)	#10002 (#2002)	#13002	#12002
...	...	...	...	...
199	#11199 (#2399)	#10199 (#2199)	#13199	#12199
200	#11200 (#2400)	#10201 (#2200)	#13200	#12200
201	#11201	#10201	#13201	#12201
...	...	...	...	...
399	#11399	#10399	#13399	#12399
400	#11400	#10400	#13400	#12400

在 FANUC 0i 系统中, 刀具补偿分为几何补偿和磨损补偿, 而且长度补偿和半径补偿也是分开的。刀具补偿号可达 400 个, 换句话说, 理论上系统支持控制容量高达 400 的刀具。

当刀具补偿号小于等于 200 时（一般情况也确实如此），刀具长度补偿（H）也可使用#2201~#2400。例如：假设一把 $\phi 10\text{mm}$ 的立铣刀，将其视为 10 号刀。若在加工中心上，是有非常明确和实在的含义，即应将此铣刀放置在刀库的 10 号位上；对于随机换刀的机床，刀号和刀库位置不一定对应；若在数控铣床上，尽管只能人工换刀，但为了使刀具管理和工艺管理更合理、更有序，同样也可效仿加工中心那样操作，即在系统中记录下相关的刀具补偿号和刀具补偿值，唯一的差别是没有刀库及自动换刀功能，而是需要人工换刀。

刀具长度补偿（H）的两项补偿值在 Z 方向对刀完成后一般不再需要特别处理，而编程时主要涉及其刀具半径补偿（D）的两项补偿值，从思路的条理性出发，显然适宜使几何补偿值等于刀具半径，即#13010=5.0，可以视为对刀具的识别，而设置和调整磨损补偿值（#12010）则可以视为对尺寸的控制。

在应用宏程序编写加工程序时，将会有以下形式的描述：

#20=#13010→把刀具补偿号 10（即 10 号刀，在此即是 $\phi 10\text{mm}$ 的立铣刀）的半径补偿值中的几何补偿值赋值给变量#20，在这里假设#20=5.0。

#22=#12010→把刀具补偿号 10（即 10 号刀，在此即是 $\phi 10\text{mm}$ 的立铣刀）的半径补偿值中的磨损补偿值赋值给变量#22，在这里假设#22=1.2。

3.3.3 宏程序报警

宏程序报警的系统变量见表 3-5。

表 3-5 宏程序报警的系统变量

变量号	功 能
#3000	当变量#3000 的值为 0~200 时，CNC 停止运行，且报警 在表达式后为不超过 26 个字符的报警信息 CRT 屏幕上显示报警号和报警信息，其中报警号为变量#3000 的值加上 3000

例如：#3000=1（TOOL NOT FOUND）。

报警屏幕上显示 3001 TOOL NOT FOUND 刀具未找到。

3.3.4 停止和信息显示

程序执行停止和显示信息见表 3-6。

表 3-6 程序执行停止和显示信息

变量号	功 能
#3006	在宏程序中指令“#3006=1（MESSAGE）；”时，程序执行前面程序段并停止 当最多到 26 个字符的信息由控制输入“(”和控制输出“)”括住，在同一程序段中编程时，信息被显示在外部操作信息屏幕上

3.3.5 时间信息

时间信息可以读和写，见表 3-7。

表 3-7 时间信息的系统变量

变量号	功 能
#3001	该变量为一个计时器，以 1ms 为计时单位。当电源接通时，该变量值复位为 0；当达到 2147483648ms 时，该计时器的值返回到 0
#3002	该变量为一个计时器，当循环启动灯亮时以 h 为单位计时。该计时器即使在电源断电时也保存该值；当达到 9544.371767h，该计时器的值返回到 0
#3011	该变量用于读取当前的日期（年/月/日）。年/月/日信息转换成十进制数。例如，1994 年 9 月 8 日表示为 19940908
#3012	该变量用于读取当前的时间（时/分/秒）。时/分/秒信息转换成十进制数。例如，下午 3 点 34 分 56 秒表示为 153456

3.3.6 自动控制

可以改变自动运行的控制状态，见表 3-8-1、表 3-8-2。

表 3-8-1 自动运行控制的系统变量（#3003）

#3003	单程序	辅助功能的完成
0	有效	等待
1	无效	等待
2	有效	不等待
3	无效	不等待

- 1) 当电源接通时该变量的值为 0。
- 2) 当单程序段停止无效时，即使单程序段开关设为 ON，也不执行单程序段停止。
- 3) 当不指定等待辅助功能（M、S 和 T 功能）完成时，在辅助功能完成之前，程序执行到下列程序段，而且分配完成信号 DEN 不输出。

表 3-8-2 自动运行控制的系统变量（#3004）

#3004	进给暂停	进给速度倍率	准确停止
0	有效	有效	有效
1	无效	有效	有效
2	有效	无效	有效
3	无效	无效	有效

(续)

#3004	进给暂停	进给速度倍率	准确停止
4	有效	有效	无效
5	无效	有效	无效
6	有效	无效	无效
7	无效	无效	无效

1) 当电源接通时，该变量的值为 0。

2) 当进给暂停无效时：

- ① 当进给暂停按钮被按下时，机床在单段停止方式停止。但是，当用变量 #3003 使单程序段方式无效时，单程序段停止不执行。
- ② 当进给暂停按钮压下又松开时，进给暂停灯亮，但是，机床不停止，程序继续执行，并且机床停在进给暂停有效的第一个程序段。
- 3) 当进给速度倍率无效时，总是 100%的倍率，而不管机床操作面板上的进给速度倍率开关的设置。

3.3.7 已加工的零件数

要求的零件数（目标数）和已加工的零件数（完成数）可以读和写，见表 3-9。

表 3-9 要求的零件数和已加工的零件数的系统变量

变量号	功 能
#3901	已加工的零件数（完成数）
#3902	要求的零件数（目标数）

以上数量不能用负值。

3.3.8 模态信息

正在处理的当前程序段之前的模态信息可以从系统变量中读出。FANUC 0i 模态信息的系统变量见表 3-10。

表 3-10 FANUC 0i 模态信息的系统变量

变量号	功 能
#4001	G00, G01, G02, G03, G33 (组 01)
#4002	G17, G18, G19 (组 02)
#4003	G90, G91 (组 03)
#4004	(组 04)

(续)

变量号	功 能
#4005	G94, G95 (组 05)
#4006	G20, G21 (组 06)
#4007	G40, G41, G42 (组 07)
#4008	G43, G44, G49 (组 08)
#4009	G73, G74, G76, G80-G89 (组 09)
#4010	G98, G99 (组 10)
#4011	G50, G51 (组 11)
#4012	G65, G66, G67 (组 12)
#4013	G96, G97 (组 13)
#4014	G54~G59 (组 14)
#4015	G61~G64 (组 15)
#4016	G68, G69 (组 16)
...	...
#4022	待定 (组 22)
#4102	B 代码
#4107	D 代码
#4109	F 代码
#4111	H 代码
#4113	M 代码
#4114	顺序号
#4115	程序号
#4119	S 代码
#4120	T 代码
#4130	P 代码 (现在选择的附加工件坐标系)

- 注：1. P 代码为当前选择的附加工件坐标系。
2. 当执行#1=#4002 时，在#1 中得到的值是 17、18、19。
3. 系统变量#4001~#4120 不能用于运算指令左边的项。
4. 模态信息不能写，只能读。如果阅读模态信息指定的系统变量为不能用的 G 代码时，系统则发出程序错误 P/S 报警。

3.3.9 当前位置信息

当前位置信息不能写，只能读。FANUC 0i 系统中当前位置信息的系统变量见表 3-11。

表 3-11 FANUC 0i 系统中当前位置信息的系统变量

变量号	位置信息	相关坐标系	移动时的读操作	刀具补偿值（长度、半径补偿）	
#5001	X 轴程序段终点位置（ABSIO）	工件坐标系	可以	不考虑刀具尖位置（程序指令位置）	
#5002	Y 轴程序段终点位置（ABSIO）				
#5003	Z 轴程序段终点位置（ABSIO）				
#5004	4th 轴程序段终点位置（ABSIO）				
#5021	X 轴当前点位置（ABSMT）	机床坐标系	不可以	考虑刀具基准点位置（机床坐标）	
#5022	Y 轴当前点位置（ABSMT）				
#5023	Z 轴当前位置（ABSMT）				
#5024	4th 轴当前位置（ABSMT）				
#5041	X 轴当前点位置（ABSOT）	工件坐标系			考虑刀具基准点位置（与位置的绝对坐标显示相同）
#5042	Y 轴当前点位置（ABSOT）				
#5043	Z 轴当前位置（ABSOT）				
#5044	4th 轴当前位置（ABSOT）				
#5061	X 轴跳跃信号位置（ABSKP）		可以	已考虑刀具基准点位置	
#5062	Y 轴跳跃信号位置（ABSKP）				
#5063	Z 轴跳跃信号位置（ABSKP）				
#5064	4th 轴跳跃信号位置（ABSKP）				
#5081	X 轴刀具长度补偿值		不可以	已考虑	
#5082	Y 轴刀具长度补偿值				
#5083	Z 轴刀具长度补偿值				
#5084	4th 轴刀具长度补偿值				
#5101	X 轴伺服位置补偿				已考虑
#5102	Y 轴伺服位置补偿				
#5103	Z 轴伺服位置补偿				
#5104	4th 轴伺服位置补偿				

- 注：1. ABSIO：工件坐标系中，前一程序段终点坐标值。
ABSMT：机床坐标系中，当前机床坐标位置。
ABSOT：工件坐标系中，当前坐标位置。
ABSKP：工件坐标系中，G31 程序段中跳跃信号有效的位置。
2. 在 G31（触发功能）程序段中，当触发信号接通时，刀具位置存储在变量#5061～#5064 中；当 G31 程序段中的触发信号不接通时，这些变量存储指定程序段的终点值。
3. 变量#5081～#5084 所存储的刀具长度补偿值是当前的执行值（即当前正在执行中的程序段的量），不是后面程序段的处理值。
4. 移动期间不能读取是由于缓冲（预读）功能的原因，不能读取目标指令值。

3.3.10 工件坐标系补偿值（工件零点偏移值）

用系统变量可以读和写工件零点偏移值，见表 3-12。

表 3-12 FANUC 0i 工件零点偏移值的系统变量

变量号	功 能
#5201	第 1 轴外部工件零点偏移值
...	...
#5204	第 4 轴外部工件零点偏移值
#5221	第 1 轴 G54 工件零点偏移值
...	...
#5224	第 4 轴 G54 工件零点偏移值
#5241	第 1 轴 G55 工件零点偏移值
...	...
#5244	第 4 轴 G55 工件零点偏移值
#5261	第 1 轴 G56 工件零点偏移值
...	...
#5264	第 4 轴 G56 工件零点偏移值
#5281	第 1 轴 G57 工件零点偏移值
...	...
#5284	第 4 轴 G57 工件零点偏移值
#5301	第 1 轴 G58 工件零点偏移值
...	...
#5304	第 4 轴 G58 工件零点偏移值
#5321	第 1 轴 G59 工件零点偏移值
...	...
#5324	第 4 轴 G59 工件零点偏移值
#7001	第 1 轴工件零点偏移值 (G54.1 P1)
...	...
#7004	第 4 轴工件零点偏移值 (G54.1 P1)
#7021	第 1 轴工件零点偏移值 (G54.1 P2)
...	...
#7024	第 4 轴工件零点偏移值 (G54.1 P2)
#7941	第 1 轴工件零点偏移值 (G54.1 P48)
...	...
#7944	第 4 轴工件零点偏移值 (G54.1 P48)

3.4 算术和逻辑运算

表 3-13 中列出的运算可以在变量中运行。等式右边的表达式可包含常量或由函数、运算符组成的变量。表达式中的变量#j 和#k 可以用常量赋值。等式左边的变量也可以用表达式赋值。其中算术运算主要是指加、减、乘、除、函数等，逻

辑运算可以理解为比较运算。

表 3-13 FANUC 0i 算术和逻辑运算

功 能		格 式	备 注
定义、置换		#i=#j	
算术运算	加法	#i=#j+#k	三角函数及反三角函数的数值均以度为 单位来指定，如 85°30'应表示为 85.5°
	减法	#i=#j-#k	
	乘法	#i=#j*#k	
	除法	#i=#j/#k	
	正弦	#i=SIN[#j] #i=ASIN[#j] #i=COS[#j] #i=ACOS[#j] #i=TAN[#j] #i=ATAN[#j]/[#k]	
	反正弦		
	余弦		
	反余弦		
	正切		
	反正切		
	平方根	#i=SQRT[#j]	
	绝对值	#i=ABS[#j]	
	舍入	#i=ROUND[#j]	
	指数函数	#i=EXP[#j]	
	（自然）对数	#i=LN[#j]	
	上取整	#i=FIX[#j]	
	下取整	#i=FUP[#j]	
	或	#i=#jOR#k #i=#jAND#k	
与			
异或	#i=#jXOR#k		
从 BCD 转为 BIN		#i=BIN[#j]	用于与 PMC 的信号交换
从 BIN 转为 BCD		#i=BCD[#j]	

以下是算术和逻辑运算指令的详细说明。

1. 反正弦运算#i=ASIN[#j]

1) 取值范围如下:

当参数 (No.6004#0) NAT 位设置为 0 时, 在 90°~270°范围内取值。

当参数 (No.6004#0) NAT 位设置为 1 时, 在-90°~90°范围内取值。

2) 当#j 超出-1 到 1 的范围时, 触发程序错误 P/S 报警 No.111。

3) 常数可替代变量#j。

2. 反余弦运算#i=ACOS[#j]

- 1) 取值范围: $0^{\circ} \sim 180^{\circ}$ 。
- 2) 当#j 超出-1 到 1 的范围时, 触发程序错误 P/S 报警 No.111。
- 3) 常数可替代变量#j

3. 反正切运算#i=ATAN[#i]/[#k]

- 1) 采用比值的书写方式 (可理解为对边/邻边)。
- 2) 取值范围如下:

当参数 (No.6004#0) NAT 位设置为 0 时, 取值范围为 $0^{\circ} \sim 360^{\circ}$ 。例如, 当指定#1=ATAN[-1]/[-1]时, #1=225°。

当参数 (No.6004#0) NAT 位设置为 1 时, 取值范围为 $-180^{\circ} \sim 180^{\circ}$ 。例如, 当指定#1=ATAN[-1]/[-1]时, #1=-135°。

- 3) 常数可替代变量#j。

4. 自然对数运算#i=LN[#j]

- 1) 相对误差可能大于 10^{-8} 。
- 2) 当反对数 (#j) 为 0 或小于 0 时, 触发程序错误 P/S 报警 No.111。
- 3) 常数可替代变量#j。

5. 指数函数#i=EXP[#j]

- 1) 相对误差可能大于 10^{-8} 。
- 2) 当运算结果超过 3.65×10^{47} (j 大约是 110) 时, 出现溢出并触发程序错误 P/S 报警 No.111。

- 3) 常数可替代变量#j。

6. 上取整#i=FIX[#j]和下取整#i=FUP[#j]

CNC 处理数值运算时, 无条件地舍去小数部分称为上取整; 小数部分进位到整数称为下取整 (注意与数学上的四舍五入对照)。对于负数的处理要特别小心。

例如: 假设#1=1.2, #2=-1.2。

- 1) 当执行#3=FUP[#1]时, 2.0 赋予#3。
- 2) 当执行#3=FIX[#1]时, 1.0 赋予#3。
- 3) 当执行#3=FUP[#2]时, -2.0 赋予#3。
- 4) 当执行#3=FIX[#2]时, -1.0 赋予#3。

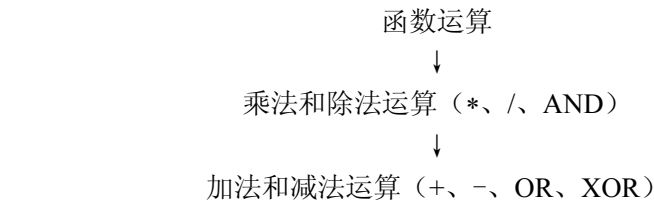
7. 算术与逻辑运算指令的缩写

程序中指令函数时, 函数名的前两个字符可以用于指定该函数。

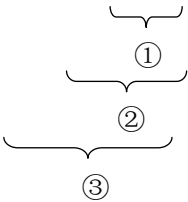
例如: ROUND→RO FIX→FI

8. 混合运算时的运算顺序

上述运算和函数可以混合运算, 即涉及运算的优先级, 其运算顺序与一般数学上的定义基本一致, 优先级顺序从高到低依次为:



例：#1=#2+#3*COS[#4]。

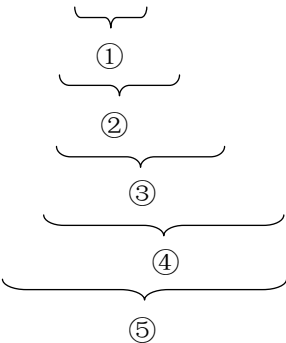


①、②、③表示运算顺序。

9. 括号嵌套

用 “[]” 可以改变运算顺序，最里层的[]优先运算。括号[]最多可以嵌套 5 级（包括函数内部使用的括号）。当超出 5 级时，触发程序错误 P/S 报警 No.118。

例：#6=COS[[[#5+#4]*#3+#2]*#1]（三重嵌套）。



①、②、③、④、⑤表示运算顺序。

10. 逻辑运算说明

逻辑运算相对于算术运算来说，更为特殊和费解，详细说明见表 3-14。

表 3-14 FANUC 0i 逻辑运算说明

运算符	功能	逻辑名	运算特点	运算实例
AND	与	逻辑乘	（相当于串联）有 0 得 0	1×1=1, 1×0=0, 0×0=0
OR	或	逻辑加	（相当于并联）有 1 得 1	1+1=1, 1+0=1, 0+0=0
XOR	异或	逻辑减	相同得 0，不同得 1	1-1=0, 1-0=1, 0-0=0, 0-1=1

11. 运算精度

同任何数学计算一样，运算的误差是不可避免的，用宏程序运算时必须考虑

用户宏程序的精度。用户宏程序处理数据的浮点格式为： $M \times 2^E$ 。

每执行一次运算，便产生一次误差，在重复计算的过程中，这些误差将累加。FANUC 0i 运算中的误差见表 3-15。

表 3-15 FANUC 0i 运算中的误差

运算	平均误差	最大误差	误差类型
$a=b*c$	1.55×10^{-10}	4.66×10^{-10}	相对误差 ^①
$a=b/c$	4.66×10^{-10}	1.88×10^{-9}	$\frac{\varepsilon}{a}$ （绝对值）
$a=\sqrt{b}$	1.24×10^{-9}	3.73×10^{-9}	
$a=b+c$ $a=b-c$	2.33×10^{-10}	5.32×10^{-10}	最小 $\frac{\varepsilon}{b}$ ， $\frac{\varepsilon}{c}$ （绝对值） ^②
$a=\text{SIN}[b]$ $a=\text{COS}[b]$	5.0×10^{-9}	1.0×10^{-8}	绝对误差 ^③ ε （绝对值）度
$a=\text{TAN}[b]/[c]$ ^④	1.8×10^{-6}	3.6×10^{-6}	

注：如果 SIN、COS 或 TAN 函数的运算结果小于 10^{-8} 或由于运算精度的限制不为 0 的话，设定参数 No.6004#1 为 1，则运算结果可视为 0。

- ① 相对误差取决于运算结果。
- ② 使用两类误差的最小者。
- ③ 绝对误差是常数，而不管运算结果。
- ④ 函数 TAN 执行 SIN/COS。

说明：

1) 加减运算。由于用户宏程序变量值的精度仅有 8 位十进制数，当在加减运算中处理非常大的数时，将得不到期望的结果。

例如：当试图把下面的值赋给变量#1 和#2 时：

#1=9876543210123.456

#2=9876543277777.777

变量值实际上已经变成：

#1=9876543200000.000

#2=9876543300000.000

此时，当编程计算#3=#2-#1 时，其结果#3 并不是期望值 67654.321，而是#3=100000.000，显然误差较大，实际计算结果其实与此还稍有误差，因为系统是以二进制执行的。

2) 逻辑运算。逻辑运算即使用条件表达式 EQ、NE、GT、GE、LT、LE 时，也可能造成误差，其情形与加减运算基本相同。

例如：IF[#1EQ#2]的运算会受到#1 和#2 误差的影响，并不总是能估算正确，要求两个值完全相同，有时不可能，由此会造成错误的判断，因此应该改用误差来限制比较稳妥，即用 IF [ABS[#1-#2] LT 0.001]代替上述语句，以避免两个变量

的误差。此时，当两个变量差值的绝对值未超过允许极限（此处为 0.001），就认为两个变量的值是相等的。

3) 三角函数运算。在三角函数运算中会发生绝对误差，它不在 10^{-8} 之内，所以注意使用三角函数后的积累误差。由于三角函数在宏程序的应用非常广泛，特别在极具数学代表性的参数方程表达上，因此必须对此保持应有的重视。

3.5 赋值与变量

赋值是指将一个数据赋予一个变量。例如： $\#1=0$ ，则表示 $\#1$ 的值是 0。其中 $\#1$ 代表变量，“ $\#$ ”是变量符号（注意：根据数控系统的不同，它的表示方法可能有差别），0 就是给变量 $\#1$ 赋的值。这里的“ $=$ ”是赋值符号，起语句定义作用。

赋值的规律有：

1) 赋值号“ $=$ ”两边内容不能随意互换，左边只能是变量，右边可以是表达式、数值或变量。

2) 一个赋值语句只能给一个变量赋值。

3) 可以多次给一个变量赋值，新变量值将取代原变量值（即最后赋的值生效）。

4) 赋值语句具有运算功能，它的一般形式为：变量=表达式。

5) 赋值表达式的运算与数学运算顺序相同。

6) 辅助功能（M 代码）的变量有最大值限制，例如，将 M30 赋值为 300 显然是不合理的。

在赋值运算中，表达式可以是变量自身与其他数据的运算结果，如： $\#1=\#1+1$ ，则表示 $\#1$ 的值为 $\#1+1$ ，这一点与数学运算是有所不同的。

需要强调的是：“ $\#1=\#1+1$ ”形式的表达式可以说是宏程序运行的“原动力”，任何宏程序几乎都离不开这种类型的赋值运算，而它偏偏与人们头脑中根深蒂固的数学上的等式概念严重偏离，因此对于初学者往往造成很大的困扰，但是，如果对计算机高级语言有一定了解的话，对此应该更易理解。

3.6 转移和循环

在程序中，使用 GOTO 语句和 IF 语句可以改变程序的流向。有三种转移和循环操作可供使用。

转移和循环 { GOTO 语句→无条件转移
IF 语句→条件转移，格式为：IF...THEN...
WHILE 语句→当...时循环

3.6.1 无条件转移（GOTO 语句）

转移（跳转）到标有顺序号 n（即俗称的行号）的程序段。当指定 1~99999 以外的顺序号时，会触发 P/S 报警 No.128。其格式为：

GOTO n; n 为顺序号（1~99999）

例如：GOTO 99，即转移至第 99 行。

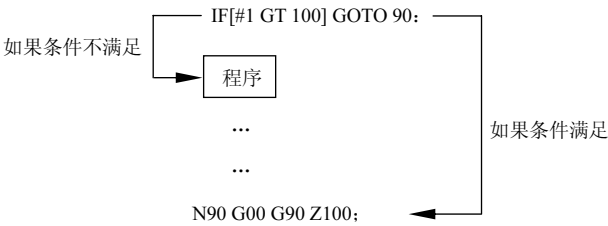
3.6.2 条件转移（IF 语句）

IF 之后指定条件表达式。

1. IF[<条件表达式>] GOTO n

表示如果指定的条件表达式满足时，则转移（跳转）到标有顺序号 n（即俗称的行号）的程序段。如果不满足指定的条件表达式，则顺序执行下一个程序段。

例如：变量#1 的值大于 100，则转移（跳转）到顺序号为 N90 的程序段。



2. IF[<条件表达式>] THEN

如果指定的条件表达式满足时，则执行预先指定的宏程序语句，而且只执行一个宏程序语句。

IF [#1 EQ #2] THEN #3=10: 如果#1 和#2 的值相同，10 赋值给#3。

说明：

1) 条件表达式必须包括运算符。运算符插在两个变量中间或变量和常量中间，并且用“[]”封闭。表达式可以替代变量。

2) 运算符由两个字母组成（见表 3-16），用于两个值的比较，以决定它们是相等还是一个值小于或大于另一个值。注意，不能使用不等号。

典型程序示例：下面的程序为计算数值 1~100 的累加总和。

O8000;

表 3-16 运算符

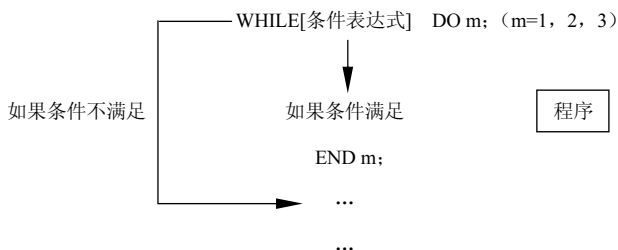
运算符	含义	英文注释
EQ	等于 (=)	Equal
NE	不等于 (≠)	Not Equal
GT	大于 (>)	Great Than
GE	大于或等于 (≥)	Great Than or Equal
LT	小于 (<)	Less Than
LE	小于或等于 (≤)	Less Than or Equal

#1=0; 存储和数变量的初值
 #2=1; 被加数变量的初值
 N5 IF [#2 GT 100] GOTO 99; 当被加数大于 100 时转移到 N99
 #1=#1+#2; 计算和数
 #2=#2+#1; 下一个被加数
 GOTO 5; 转到 N5
 N99 M30; 程序结束

3.6.3 循环（WHILE 语句）

在 WHILE 后指定一个条件表达式,当指定条件满足时,则执行从 DO 到 END 之间的程序,否则,转到 END 后的程序段。

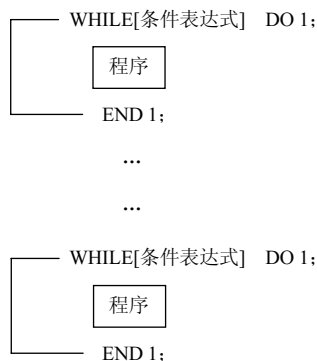
DO 后面的号是指定程序执行范围的标号,标号值为 1、2、3。如果使用了 1、2、3 以外的值,会触发 P/S 报警 No.126。



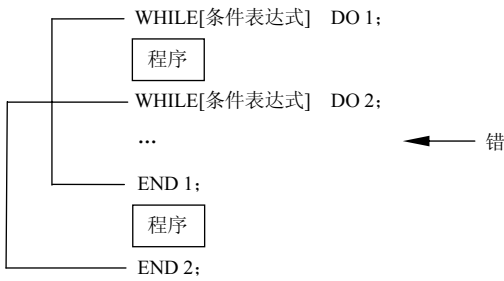
1. 嵌套

在 DO-END 循环中的标号(1~3)可根据需要多次使用。但是需要注意的是,无论怎样多次使用,标号永远限制在 1、2、3。此外,当程序有交叉重复循环(DO 范围的重叠)时,会触发 P/S 报警 No.124。以下为关于嵌套的详细说明。

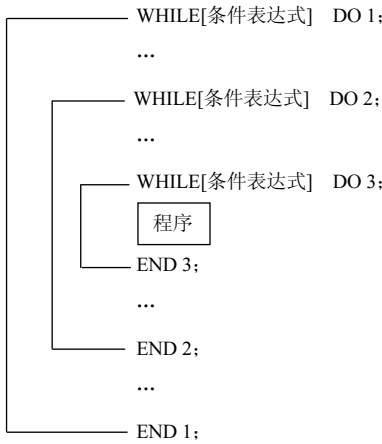
1) 标号(1~3)可以根据需要多次使用。



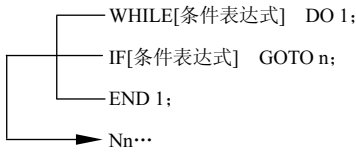
2) DO 的范围不能交叉。



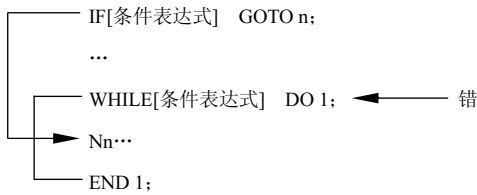
3) DO 循环可以三重嵌套。



4) (条件) 转移可以跳出循环的外边。



5) (条件) 转移不能进入循环区内，注意与上述第 4) 点对照。



2. 关于循环 (WHILE 语句) 的其他说明

1) DO m 和 END m 必须成对使用，而且 DO m 一定要在 END m 指令之前。用识别号 m 来识别。

2) 当指定 DO 而没有指定 WHILE 语句时，将产生从 DO 到 END 之间的无限循环。

3) 在使用 EQ 或 NE 的条件表达式中, 值为空和值为零将会有不同的效果。而在其他形式的条件表达式中, 空即被当作零。

4) 条件转移 (IF 语句) 和循环 (WHILE 语句) 从逻辑关系上说, 两者不过是从正反两个方面描述同一件事情; 从实现的功能上说, 两者具有相当程度的相互替代性; 从具体的用法和使用的限制上说, 条件转移 (IF 语句) 受到系统的限制相对更少, 使用更灵活。

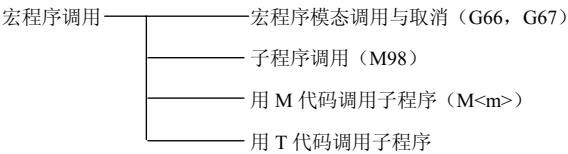
5) 当在 GOTO 语句 (无论是无条件转移的 GOTO 语句, 还是 “IF…GOTO” 形式的条件转移 GOTO 语句) 中有标号转移的语句时, 系统将进行顺序号检索。一般来说数控系统执行反向检索的时间要比正向检索长, 因为系统通常先正向搜索到程序结束, 再返回程序开头进行搜索, 所以花费的时间要多。因此, 用 WHILE 语句实现循环可减少处理时间。

在宏程序的应用中, 优先考虑的应该是数学表达是否正确, 思路是否简洁, 逻辑是否严密, 至于具体选择何种语句来实现, 则不必拘泥。事实上, 从实践经验来看, 这里所讨论的处理时间在实际应用中差别并不明显, 而且, 从宏程序的学习和掌握技巧来看, 似乎 “IF…GOTO” 形式的条件转移语句比 GOTO 语句相对更容易理解和掌握, 特别是对于初学者而言。

第 4 章 用户宏功能

4.1 用户宏程序调用指令 A

用户宏指令是调用用户宏程序的指令，用户宏程序功能 A 用以下方法调用宏程序：



4.1.1 宏程序模态调用与取消 (G66、G67)

指令格式为：G66 P <p>;

式中 P 为调用的宏程序本体程序号。用上述指令时，系统为宏程序模态调用方式，即其后的每个程序每执行一次，便调用一次 P 指令的宏程序，并且在其后的各程序中都可以指定自变量。

取消宏程序模态调用指令 G67，即取消宏程序模态调用指令 G66。G66 和 G67 应该成对使用。

4.1.2 子程序调用 (M98)

指令格式为：M98 P <p>;

式中 P 为调用的宏程序本体程序号。用上述指令，可调用 P 指定的宏程序本体。

4.1.3 用 M 代码调用子程序

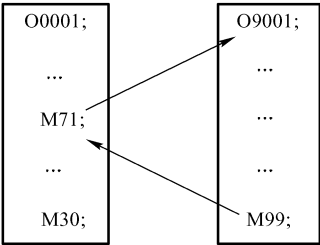
可用 M <m>代码代替 M98 P<p>。在参数 No.6071～No.6079 中设定调用子程序的 M 代码<m>，可用与子程序调用 (M98) 相同的方法调用子程序，表 4-1 为 FANUC 0i 参数、M 代码与子程序号之间的对应关系。

<m>值范围可从 03～97 中选取，其中 30 和不能进入缓冲寄存器的 M 代码除外。

表 4-1 FANUC 0i 参数、M 代码与子程序号之间的对应关系

参数号	M 代码<m>	被调用的用户子程序号<p>
6071	m1	O9001
6072	m2	O9002
6073	m3	O9003
6074	m4	O9004
6075	m5	O9005
6076	m6	O9006
6077	m7	O9007
6078	m8	O9008
6079	m9	O9009

例如：假设在系统中将 No.6071 参数设置为 71（为了识别的方便性和条理性，建议将 m1~m9 依次设置为 71~79），则：M71=M98 P9001。



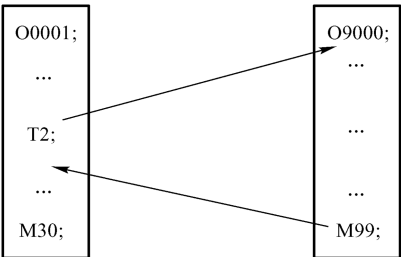
参数 No.6071=71

说明：

- 1) 不允许自变量赋值。
- 2) 在宏程序中调用的 M 代码被处理为普通的 M 代码。

4.1.4 用 T 代码调用子程序

设置参数 No.6001 的#5 位 TCS=1 时，可用 T <t>代码代替 M98 P9000。在加工程序中指定的 T 代码<t>赋值到（存储）公共变量#149 中。



参数 No.6001 的#5 位 TCS=1

公共变量#149=22

说明:

1) 设置参数 No.6001 的#5 位 TCS=1 时, 可用 T<t>代码代替 M98 P9000。在加工程序中指定的 T 代码<t>赋值到 (存储) 公共变量#149 中。

2) 在用 G 代码调用的宏程序或用 M、T 代码作为子程序调用的程序中, 不能用一个 T 代码调用多个子程序。这种宏程序或程序中的 T 代码被处理为普通的 T 代码。

4.2 用户宏程序本体

4.2.1 用户宏程序本体的结构

在用户宏程序本体中, 可以使用普通 NC 指令、采用变量的 NC 指令、计算指令和转换指令。

用户宏程序本体的结构, 以字母 O 后的程序号开始, 用 M99 结束。

例如:

O×××××; 程序号

G65 H05……; 运算指令

G90 G00 X#110 Y#120; 使用变量的 NC 指令

…;

G65 H82……; 转移指令

M99; 用户宏程序本体结束

4.2.2 变量的表示和引用

用变量可以指定用户宏程序本体的地址值。当调用或执行宏程序时, 通过计算变量, 可以指定一个变量, 使宏程序更加灵活, 用途更加广泛。

用变量可以替代地址后面的具体数值。

例如: F#110, 当#110=100 时, 相当于 F100。

Z-#120, 当#120=150.0 时, 相当于 Z-150.0。

G#130, 当#130=2 时, 相当于 G2 (即 G02)。

当用变量替代变量号时, 不能表示为 “# #1000” 或 “#[#100]”, 而应写成 “#9100”, 即用 “9” 替代后面的 “#” 表示替换的变量号。

例如: 若#100=#110, #110=-400, 则 X#9100 表示 X-400, 而 X-#9100 则表示 X400。

注意：地址 O 和 N 不能引用变量。指令值不能超过各地址的最大指令值。若 #130=300，则 G#130 超过了最大指令值。

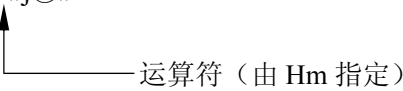
4.2.3 变量的种类

在用户宏程序功能 A 中，使用的变量是公共变量和系统变量，这点一定要注意，相关变量的定义和特点参见前文所述。

4.2.4 宏程序的运算和控制指令

指令格式：G65 Hm P#i Q#j R#k

式中 m 为 01~99，表示宏程序功能；#i 为存储运算结果的变量号；#j 为进行运算的变量号，也可以是常数；#k 为进行运算的变量号，也可以是常数。

意义：#i=#j①#k


注意：变量值不能带小数，与各地址不带小数时所表示的意义相同（参数 No.3401 的#0 位 DPI=O，最小输入单位 0.001mm 及 0.001°）

例如：若#100=20，以 0.001mm 为单位输入时，X#100 为 X0.01mm（20×0.001mm=0.020mm）；

若#100=200，以 0.001°角度为单位输入时，#100 为 0.10°（200×0.001°=0.20°），G65Hm 宏指令见表 4-2。

表 4-2 FANUC 0i G65Hm 宏指令表（宏程序的运算与控制指令）

G65Hm	功 能	数学定义
G65H01	定义、置换	#i=#j
G65H02	加法	#i=#j+#k
G65H03	减法	#i=#j-#k
G65H04	乘法	#i=#j*#k
G65H05	除法	#i=#j÷#k
G65H11	逻辑加	#i=#j AND #k
G65H12	逻辑乘	#i=#jOR#k
G65H13	异或	#i=#jXOR#k
G65H21	开平方	#i=√#j
G65H22	绝对值	#i= #j
G65H23	剩余数	#i=#j-TRUNC(#j/#k)*#k TRUNC：小数部分舍去

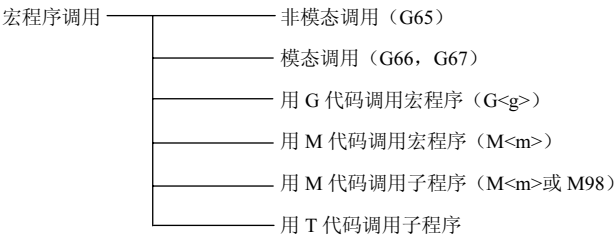
(续)

G65Hm	功 能	数学定义
G65H24	变成二进制	$\#i = \text{BIN}(\#j)$
G65H25	变成十进制	$\#i = \text{BCD}(\#j)$
G65H26	复合除运算	$\#i = (\#j * \#k) \div \#k$
G65H27	复合平方根 1	$\#i = \sqrt{\#j * \#j + \#k + \#k}$
G65H28	复合平方根 2	$\#i = \sqrt{\#j * \#j - \#k + \#k}$
G65H31	正弦	$\#i = \#j * \sin(\#k)$
G65H32	余弦	$\#i = \#j * \cos(\#k)$
G65H33	正切	$\#i = \#j * \tan(\#k)$
G65H34	反正切	$\#i = \arctan(\#j / \#k)$
G65H80	无条件转移	GOTO n
G65H81	条件转移 1	2F $\#j = \#k$ GOTO n
G65H82	条件转移 2	2F $\#j \neq \#k$ GOTO n
G65H83	条件转移 3	2F $\#j > \#k$ GOTO n
G65H84	条件转移 4	2F $\#j < \#k$ GOTO n
G65H85	条件转移 5	2F $\#j \geq \#k$ GOTO n
G65H86	条件转移 6	2F $\#j \leq \#k$ GOTO n
G65H99	P/S 报警	报警号为 500+n

从 G65Hm 指令的用法定义可以看出，用户宏程序功能 A 的使用是非常烦琐且极不直观，因此，在真正的生产实践中，用户宏程序功能 A 的使用机会是很少的。

4.3 用户宏程序调用指令 B

用户宏指令是调用用户宏程序的指令，用户宏程序功能 B 用以下方法调用宏程序：



首先说明用户宏程序调用（G65）与子程序调用（M98）之间的差别：

1) G65 可以进行自变量赋值，即指定自变量（数据传送到宏程序），M98 则不能。

2) 当 M98 程序段包含另一个 NC 指令（例如，G01 X200.0 M98 P<p>）时，在执行完这种含有非 N、P 或 L 的指令后可调用（或转移到）子程序。相反，G65 则只能无条件地调用宏程序。

3) 当 M98 程序段包含有 O、N、P、L 以外地址的 NC 指令（例如 G01 X200.0 M98 P<p>）时，在单程序段方式中，可以单程序段停止（即停机）。相反，G65 则不行（即不停机）。

4) G65 改变局部变量的级别，M98 不改变局部变量的级别。

4.3.1 宏程序非模态调用（G65）

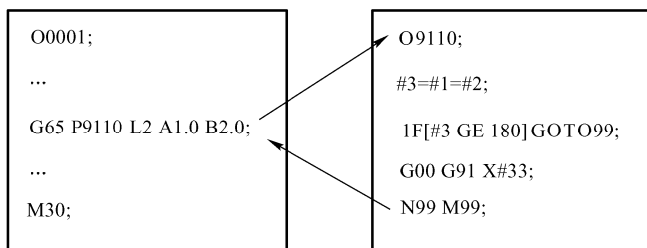
当指定 G65 时，调用以地址 P 指定的用户宏程序，数据（自变量）能传递到用户宏程序中，指令格式如下。

G65 P<p> L<l> <自变量赋值>;

<p>：要调用的程序号；

<l>：重复次数（默认值为 1）；

<自变量赋值>：传递到宏程序的数据。



1. 调用说明

1) 在 G65 之后，用地址 P 指定用户宏程序的程序号。

2) 任何自变量前必须指定 G65。

3) 当要求重复时，在地址 L 后指定 1~9999 的重复次数，省略 L 值时，默认 L 值等于 1。

4) 使用自变量指定（赋值），其值被赋值给宏程序中相应的局部变量。

2. 自变量指定（赋值）

自变量指定又可称之为自变量赋值（以下统一采用该叫法），即若要向用户宏程序本体传递数据时，须由自变量赋值来指定，其值可以有符号和小数点，且与地址无关。

这里使用的是局部变量（#1～#33 共有 33 个），与其对应的自变量赋值共有两种类型：

自变量赋值 I：用英文字母后加数值进行赋值，除了 G、L、O、N 和 P 之外，其余所有 21 个英文字母都可以给自变量赋值，每个字母赋值一次，从 A、B、C、D、到 X、Y、Z，赋值不必按字母顺序进行，但使用 I、J、K 时，必须按字母顺序指定（赋值），不赋值的地址可以省略。

自变量赋值 II：与自变量指定赋值 I 类似，也是用英文字母后加数值进行赋值，但只用了 A、B、C 和 I、J、K 这 6 个字母，具体用法是：除了 A、B、C 之外，还用 10 组 I、J、K 来对自变量进行赋值，在这里 I、J、K 是分组定义的，同组的 I、J、K 必须按字母顺序指定，不赋值的地址可以省略。

自变量赋值 I 和自变量赋值 II 与用户宏程序本体中局部变量的对应关系见表 4-3、表 4-4。

表 4-3 FANUC 0i 第 I 类自变量赋值

地址	变量号	地址	变量号	地址	变量号
A	#1	I	#4	T	#20
B	#2	J	#5	U	#21
C	#3	K	#6	V	#22
D	#7	M	#13	W	#23
E	#8	Q	#17	X	#24
F	#9	R	#18	Y	#25
H	#11	S	#19	Z	#26

表 4-4 FANUC 0i 第 II 类自变量赋值

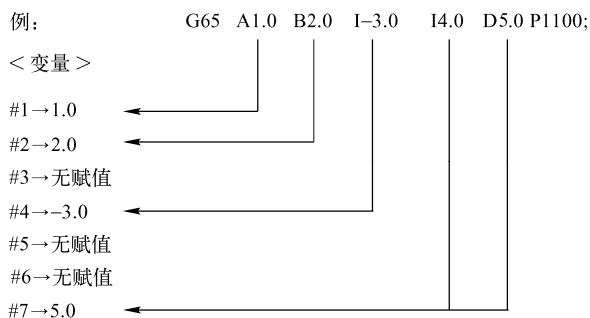
地址	变量号	地址	变量号	地址	变量号
A	#1	K ₃	#12	J ₇	#23
B	#2	I ₄	#13	K ₇	#24
C	#3	J ₄	#14	I ₈	#25
I ₁	#4	K ₄	#15	J ₈	#26
J ₁	#5	I ₅	#16	K ₈	#27
K ₁	#6	J ₅	#17	I ₉	#28
I ₂	#7	K ₅	#18	J ₉	#29
J ₂	#8	I ₆	#19	K ₉	#30
K ₂	#9	J ₆	#20	I ₁₀	#31
I ₃	#10	K ₆	#21	J ₁₀	#32
J ₃	#11	I ₇	#22	K ₁₀	#33

注意：对于自变量赋值Ⅱ，表 4-4 中 I、J、K 的下标用于确定自变量赋值的顺序，在实际编程中不写（也无法写，语法上无法表达）。

3. 自变量赋值的其他说明

1) 自变量赋值Ⅰ、Ⅱ的混合使用。CNC 内部自动识别自变量赋值Ⅰ和Ⅱ。

赋值Ⅰ和Ⅱ混合赋值，较后赋值的自变量类型有效（以从左到右书写的顺序为准，左为先，右为后）。



该例中，I4.0 和 D5.0 都给变量#7 赋值，但后者 D5.0 有效。

由此可以看出，自变量赋值Ⅱ用 10 组 I、J、K 来对自变量进行赋值，在上述表 4-4 中似乎可以通过 I、J、K 的下标很容易识别地址和变量的关系，但实际上存在实际编程中无法输入下标的问题，尽管自变量值Ⅱ“充分利用资源”，可以对 #1~#33 全部 33 个局部变量进行赋值，但是在实际编程时要分清是哪一组 I、J、K，又是第几个 I 或 J 或 K，是一件非常麻烦的事。如果再让自变量赋值Ⅰ和自变量赋值Ⅱ混合使用，那就更是烦上加烦。

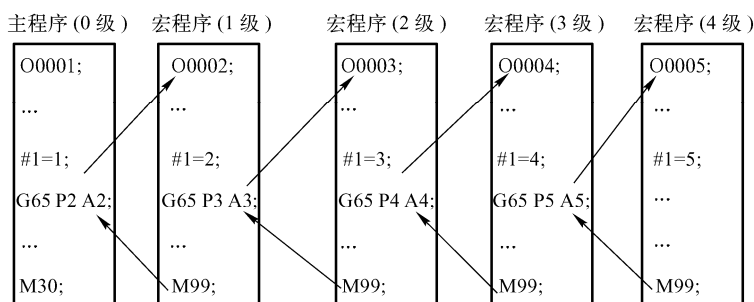
相反，如果只用自变量赋值Ⅰ进行赋值，由于地址和变量是一一对应的关系，混淆和出错的机会相当小，尽管只有 21 个英文字母可以给自变量赋值，但是 95% 以上的编程工作再复杂也不会出现超过 21 个变量的情况。因此，在此建议在实际编程时，使用自变量赋值Ⅰ进行赋值。

2) 小数点的问题。没有小数点的自变量数据的单位为各地址的最小设定单位。传递的没有小数点的自变量的值将根据机床实际的系统配置而定。因此建议在宏程序调用中一律使用小数点，既可避免无谓的差错，也可使程序对机床及系统的兼容性好。

3) 调用嵌套。调用可以四级嵌套，包括非模态调用（G65）和模态调用（G66），但不包括子程序调用（M98）。

4) 局部变量的级别。局部变量嵌套从 0 到 4 级，主程序是 0 级。用 G65 或 G66 调用宏程序，每调用一次（2、3、4 级），局部变量级别加 1，而前一级的局部变量值保存在 CNC 中，即每级局部变量（1、2、3 级）被保存，下一级的局部变量（2、3、4 级）被准备，可以进行自变量赋值。

当宏程序执行 M99 时，控制返回到调用的程序，此时，局部变量级别减 1，并恢复宏程序调用时保存的局部变量值，即上一级被储存的局部变量被恢复，如同它被储存一样，而下一级的局部变量被清除。



4.3.2 宏程序模态调用与取消 (G66、G67)

当指定 G66 时，则指定宏程序模态调用，即指定沿移动轴移动的程序段后调用宏程序，G67 取消宏程序模态调用。指令格式与非模态调用 (G65) 相似。

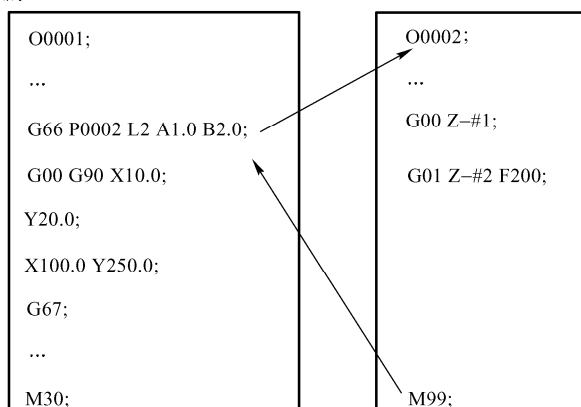
G66 P<p> L<l> <自变量赋值>;

<p>: 要调用的程序号;

<l>: 重复次数 (默认值为 1);

<自变量赋值>: 传递到宏程序的数据。

例如:



说明:

1) 在 G66 之后，用地址 P 指定用户宏程序的程序号。

2) 任何自变量前必须指定 G66。

3) 当要求重复时，在地址 L 后指定 1~9999 的重复次数，省略 L 值时，默认 L 值等于 1。

- 4) 与非模态调用（G65）相同，使用自变量指定（赋值），其值被赋值给宏程序中相应的局部变量。
- 5) 指定 G67 时，取消 G66，即其后面的程序段不再执行宏程序模态调用。G66 和 G67 应该成对使用。
- 6) 可以调用四级嵌套，包括非模态调用（G65）和模态调用（G66）。但不包括子程序调用（M98）。
- 7) 在模态调用期间，指定另一个 G66 代码，可以嵌套模态调用。
- 8) 限制：
 - ① 在 G66 程序段中，不能调用多个宏程序。
 - ② 在只有诸如辅助功能（M 代码），但无移动指令的程序段中不能调用宏程序。
 - ③ 局部变量（自变量）只能在 G66 程序段中指定，注意，每次执行模态调用时，不再设定局部变量。

4.3.3 用 G 代码调用宏程序（G <g>）

可用 G <g>代码代替 G65 P <p>，即：
G <g> <自变量赋值>=G65 P <p> <自变量赋值>。

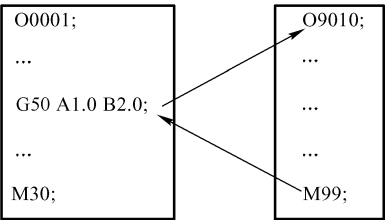
在参数 No.6050～No.6059 中设定调用宏程序的 G 代码 <g>，该代号用与非模态调用（G65）同样的方法调用宏程序。对应关系见表 4-5。

表 4-5 FANUC 0i 参数、G 代码与宏程序号之间的对应关系

参数号	G 代码 <g>	被调用的用户宏程序号<p>
6050	g1	O9010
6051	g2	O9011
6052	g3	O9012
6053	g4	O9013
6054	g5	O9014
6055	g6	O9015
6056	g7	O9016
6057	g8	O9017
6058	g9	O9018
6059	g10	O9019

<g>值范围为 1～255（65～67 除外）。

例如：假设在系统中将 No.6050 参数设置为 50（为了识别的方便性和条理性，强烈建议将 g1～g10 依次设置为 50～59），则 G50 即为 G65 P9010。



参数 No.6050=50

说明：

- 1) 与非模态调用（G65）完全一样，地址 L 可以指定 1～9999 的重复次数。
- 2) 自变量赋值与非模态调用（G65）完全一样。

3) 在用 G 代码调用的程序中，不能用一个 G 代码调用多个宏程序。这种程序中的 G 代码被处理为普通的 G 代码。

在用 M 或 T 代码作为子程序调用的程序中，不能用一个 G 代码调用多个宏程序。这种程序中的 G 代码也被处理为普通的 G 代码。

4.3.4 用 M 代码调用宏程序（M <m>）

可用 M <m>代码代替 G65 P<p>，即：

M <m> <自变量赋值>=G65 P<p> <自变量赋值>。

在参数 No.6080～No.6089 中设定调用宏程序的 M 代码<m>，该代码用与非模态调用（G65）同样的方法调用宏程序。对应关系见表 4-6。

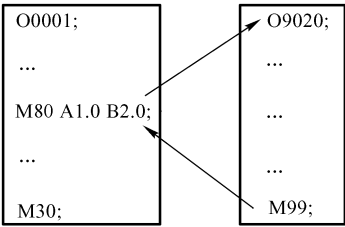
表 4-6 FANUC 0i 参数、M 代码与宏程序号之间的对应关系

参数号	M 代码<m>	被调用的用户宏程序号<p>
6080	m1	O9020
6081	m2	O9021
6082	m3	O9022
6083	m4	O9023
6084	m5	O9024
6085	m6	O9025
6086	m7	O9026
6087	m8	O9027
6088	m9	O9028
6089	m10	O9029

<m>值范围为 6～255。

例如：假设在系统中将 No.6080 参数设置为 80（为了识别的方便性和条理性，

强烈建议将 m1~m10 依次设置为 80~89)，则 M80 即为 G65 P9020。



参数 No.6080=80

说明：

- 1) 与非模态调用（G65）完全一样，地址 L 可以指定 1~9999 的重复次数。
- 2) 自变量赋值与非模态调用（G65）完全一样。
- 3) 限制：

① 调用宏程序的 M 代码必须在程序段的开头指定。

② 在用 G 代码调用的宏程序或用 M、T 代码作为子程序调用的程序中，不能用一个 M 代码调用多个宏程序。这种宏程序或程序中的 M 代码被处理为普通的 M 代码。

4.3.5 用 M 代码调用子程序

用 M 代码调用子程序与前述的用户宏程序功能 A 相应章节基本相似，可用 M <m>代码代替 M98 P <p>。

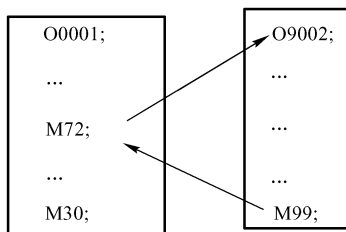
在参数 No.6071~No.6079 中设定调用子程序（宏程序）的 M 代码<m>，该代码用与子程序调用（M98）相同的方法调用子程序（宏程序）。对应关系见表 4-7。

表 4-7 FANUC 0i 参数、M 代码与子程序号之间的对应关系

参数号	M 代码<m>	被调用的用户宏程序号<p>
6071	m1	O9001
6072	m2	O9002
6073	m3	O9003
6074	m4	O9004
6075	m5	O9005
6076	m6	O9006
6077	m7	O9007
6078	m8	O9008
6079	m9	O9009

<m>值可从 03~97 中选取，但是 30 和不能进入缓冲寄存器的 M 代码除外。

例如：假设在系统中将 No.6072 参数设置为 72（为了识别的方便性和条理性，强烈建议将 m1~m9 依次设置为 71~79），则 M72 即为 M98 P9002



说明：

1) 与非模态调用（G65）完全一样，地址 L 可以指定 1~9999 的重复次数。

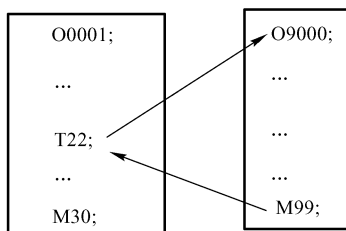
2) 不允许自变量赋值。

3) 在用 G 代码调用的宏程序或用 M、T 代码作为子程序调用的程序中，不能用一个 M 代码调用多个子程序。这种宏程序或程序中的 M 代码被处理为普通的 M 代码。

4.3.6 用 T 代码调用子程序

可用 T<t>代码代替 M98 P<p>。

在参数中设定调用子程序（宏程序）的 T 代码<t>，该代码用与子程序调用（M98）相同的方法调用子程序（宏程序），与前述的用户宏程序功能 A 相应章节基本相似。



参数 No.6001 的#5 位 TCS=1

公共变量#149=22

说明：

1) 调用：设置参数 No.6001 的#5 位 TCS=1 时，可用 T<t>代码代替 M98 P9000。在加工程序中指定的 T 代码<t>赋值到（存储）公共变量#149 中。

2) 限制：在用 G 代码调用的宏程序或用 M、T 代码作为子程序调用的程序中，不能用一个 T 代码调用多个子程序。这种宏程序或程序中的 T 代码被处理为普通的 T 代码。

4.4 宏程序语句和 NC 语句

4.4.1 宏程序语句和 NC 语句的定义

在宏程序中，可以把程序段分为两种语句，一种为宏程序语句，一种为 NC 语句。以下类型的程序段均属宏程序语句：

- 1) 包含算术或逻辑运算(=)的程序段。
- 2) 包含控制语句(例如，GOTO，DO~END)的程序段。
- 3) 包含宏程序调用指令(例如，用 G65，G66，G67 或其他 G、M 代码调用宏程序)的程序段。

除了宏程序语句以外的任何程序段都是 NC 语句。

4.4.2 宏程序语句和 NC 语句的异同

1. 宏程序语句与 NC 语句的区别

1) 宏程序语句即使置于单程序段运行方式，机床也不停止运行。但是，当参数 No.6000#5SBM 设定为 1 时，在单程序段方式中也执行单程序段停止（这只在调试时才使用）。

2) 在刀具半径补偿方式中宏程序语句段不作为不移动程序段处理。

2. 与宏程序语句有相同功能的 NC 语句

1) NC 语句含有子程序调用程序段，包括 M98、M 和 T 代码调用子程序的指令，但只包括子程序调用指令和地址 O、N、P、L。

2) NC 语句含 M99 的程序段，但只包括地址 O、N、P、L。

4.4.3 宏程序语句的处理

为了平滑加工，CNC 会预读下一个要执行的 NC 语句，这种运行称为缓冲。

在刀具半径补偿方式(G41，G42)中，CNC 为了找到交点会提前预读 2 或 3 个程序段的 NC 语句。

算术表达式和条件转移的宏程序语句在它们被读进缓冲寄存器后立即被处理。

CNC 不预读以下三种类型的程序段：包含 M00，M01，M02 或 M30 的程序段；包含由参数 No.3411~No.3420 设置的禁止缓冲的 M 代码的程序段；包含 G31 的程序段。

4.4.4 用户宏程序的存储

用户宏程序与子程序相似。可用与子程序同样的方法进行存储和编程，存储

容量由子程序和宏程序的总容量确定。

4.5 用户宏程序的使用限制

1. MDI 运行

在 MDI 方式中，不可以指定宏程序，但可进行下列操作：调用子程序；调用一个宏程序，但该宏程序在自动运行状态下不能调用另一个宏程序。

2. 顺序号检索

用户宏程序不能检索顺序号。

3. 单程序段

1) 除了包含宏程序调用指令、运算指令和控制指令的程序段之外，可以执行一个程序段作为一个单程序的停止（在宏程序中），换言之，即使宏程序在单程序段方式下正在执行，程序段也能停止。

2) 包含宏程序调用指令（G65/G66）的程序段中即使单程序段方式时也不能停止。

3) 当设定参数 SBM（参数 No.6000 的#5 位）为 1 时，包含算术运算指令和控制指令的程序段可以停止（即单程序段停止）。该功能主要用于检查和调试用户宏程序本体。

注意：在刀具半径补偿方式中，当宏程序语句中出现单程序段停止时，该语句被认为不包含移动的程序段，并且在某些情况下，不能执行正确的补偿（严格地说，该程序段被当作指定移动距离为 0 的移动。）

4. 使用任选程序段跳过（跳跃功能）

在<表达式>中间出现的“/”符号（即在算术表达式的右边，封闭在[]中）被认为是除法运算符，而不作为任选程序段跳过代码。

5. 在 EDIT 方式下的运行

1) 设定参数 NE8（参数 No.3202 的#0 位）和 NE9（参数 No.3202 的#4 位）为 1 时，可对程序号为 8000~8999 和 9000~9999 的用户宏程序和子程序进行保护。

2) 当存储器全清时（电源接通时，同时按下 **REST** 和 **DELETE** 键），存储器的全部内容包括宏程序（子程序）将被清除。

6. 复位

1) 复位后，所有局部变量和从#100~#149 的公共变量被清除为空值。

2) 设定参数 CLV（参数 No.6001 的#7 位）和 CCV（参数 No.6001 的#6 位）为 1 时，它们可以不被清除（这取决于机床制造厂）。

3) 复位不清除系统变量#1000~#1133。

4) 复位可清除任何用户宏程序和子程序的调用状态及 DO 状态并返回到主程序。

7. 程序再启动的显示

和 M98 一样，子程序调用使用的 M、T 代码不显示。

8. 进给暂停

在宏程序语句的执行期间，且进给暂停有效时，在宏程序执行完成之后机床停止。当复位或出现报警时，机床停止。

9. <表达式>中可以使用的常数值

表达式中可以使用的常数值为在 “+0.0000001 ~ +99999999 ” 以及 “-99999999 ~ -0.0000001 ” 范围内的 8 位十进制数，如果超过这个范围，会触发 P/S 报警 No.003。

4.6 外部输出指令

除了标准的用户宏指令外，还可用下面的宏程序指令。它们被称作外部输出指令。

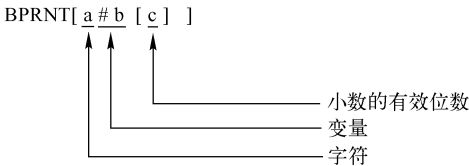
- BPRNT
- DPRNT
- POPEN
- PCLOS

通过阅读机/穿孔机接口，用这些指令输出变量值和字符。

4.6.1 打开指令 POPEN

POPEN 建立与外部输入/输出设备的链接。必须在数据输出指令的顺序之前指定。根据该指令 CNC 输出 DC2 控制代码。

4.6.2 数据输出指令 BPRNT



BPRNT 指令以二进制输出字符和变量值。

- 1) 指定的字符根据输出时的设定数据（ISO）转换成代码。

可指定的字符如下：

字母 A 到 Z

数字

特殊字符 (*, /, +, -) 等

星号 (*) 用空格输出。

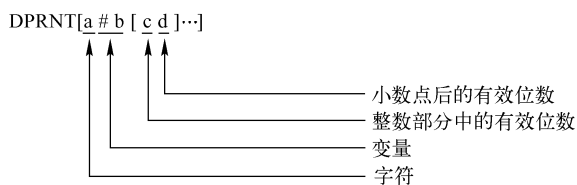
2) 所有变量都以小数点存储。

指定一个变量时，变量后要用括号。指定小数的有效位数。变量值处理成为两个字节（32 位）数据，包括小数位。输出为二进制数据，从最高字节开始输出。

3) 当指定的数据输出时，根据设定代码（ISO）输出 EOB 代码。

4) 空变量认为是 0。

4.6.3 数据输出指令 DPRNT



DPRNT 指令根据 SETTING（ISO）的代码设定，输出字符和变量中的各位数值。

1) 指定的字符根据输出时的设定数据（ISO）转换成代码。

可指定的字符如下：

字母 A 到 Z

数字

特殊字符 (*, /, +, -) 等

星号 (*) 用空格输出。

2) 当指定的数据输出时，根据设定代码（ISO）输出 EOB 代码。

3) 空变量认为是 0。

4) 当输出变量时，要指定#和后面的变量号，然后，在括号中指定整数部分中的位数和小数点后的位数。

数据从高位输出，每位数都按相应的代码输出，代码由（ISO）设定。小数点也按（ISO）设定的代码输出。

每个变量必须是最多 8 位数组成的数值。当高位数是零时，如果 PRT（参

数 6001 的 1 位) 是 1 的话, 这些零不输出; 如果参数 PRT 是 0, 每当遇到 0 时输出空格代码。

当小数的有效位数不是零时, 小数部分总是输出; 如果小数的有效位数是零, 则不输出小数点。

当 PRT (参数 6001 的 1 位) 是 0 时, 输出空格代替+号表示正数; 如果参数 PRT 是 1, 不输出代码。

4.6.4 关闭指令 PCLOS

PCLOS 指令解除与外部输入/输出设备的链接。当所有数据输出指令结束时, 指定该指令。根据该指令, CNC 输出 DC4 控制代码。

4.6.5 要求的设定

指定设定 (SETTING) 数据中的通道 (I/O 通道), 根据这些数据指定阅读机/穿孔机接口的数据 (例如波特率)。

I/O 通道 0: 参数 No.101, No.102 和 No.103

I/O 通道 1: 参数 No.111, No.112 和 No.113

I/O 通道 2: 参数 No.112, No.122 和 No.123

输出装置不要指定 FANUC 磁带盒或软盘盒。

当指定 DPRNT 指令输出数据时, 用参数 PRT (参数 6001 的 1 位) 为 1 或 0 指定是否输出数据的前零。

数据行结束时输出 LF 或 CF, 用参数#6001 的第 4 位 CRO 设定。

4.7 中断型用户宏程序

当程序正在执行时, 来自机床的中断输入信号 (UNIT) 可以调用另一个程序。该功能称为中断型用户宏程序功能。以下面的格式编程中断指令。

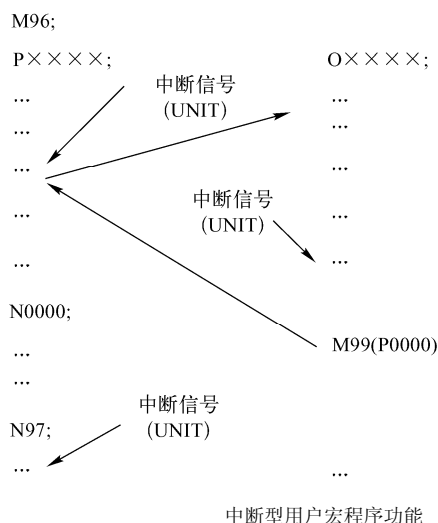
4.7.1 指令格式

M96 P0000; 用户宏程序中断有效

M97; 用户宏程序中断无效

中断型用户宏程序功能允许在程序执行期间在任一程序段调用另一个程序。这就可使程序根据随时变化的条件实时地运行。

- 1) 当检测到刀具的异常状态时, 用外部信号启动异常过程的处理。
- 2) 加工运行的顺序由另一个加工操作中中断而不取消现在的运行。
- 3) 以固定的间隔, 读取与现在加工有关的信息。



当在程序中指定 **M96 P××××** 时，由于中断信号（UNIT）的输入，中断了后面的程序运行，转为执行由 **P××××** 指令的程序。

4.7.2 指定方法说明

1. 中断条件

用户宏程序只能在程序执行期间有效。在下列条件下使用：

- 1) 存储器运行或 MDI 运行时。
- 2) STL 启动灯亮时。
- 3) 目前不处理用户宏程序中断时。

2. 指定

一般情况下，指定 **M96** 使中断信号（UNIT）有效，可以执行用户宏程序中断功能。**M97** 使信号无效。一旦指定 **M96**，中断信号（UNIT）的输入使用户宏程序中断开始。直到指定 **M97** 或 NC 复位后，中断型宏程序功能取消。在指定 **M97** 或 NC 复位以后，即使中断信号（UNIT）输入，用户宏程序中断也不执行，中断信号被忽略直到指定另一个 **M96**。指定 **M96** 后，中断信号（UNIT）生效。在 **M97** 方式中即使输入信号，也被忽略。当在 **M97** 方式中断信号输入并保持接通直到 **M96** 指定时，（用状态触发形式时）立即执行宏程序中断。当使用边沿触发形式时，即使指定 **M96**，用户宏程序也不执行。

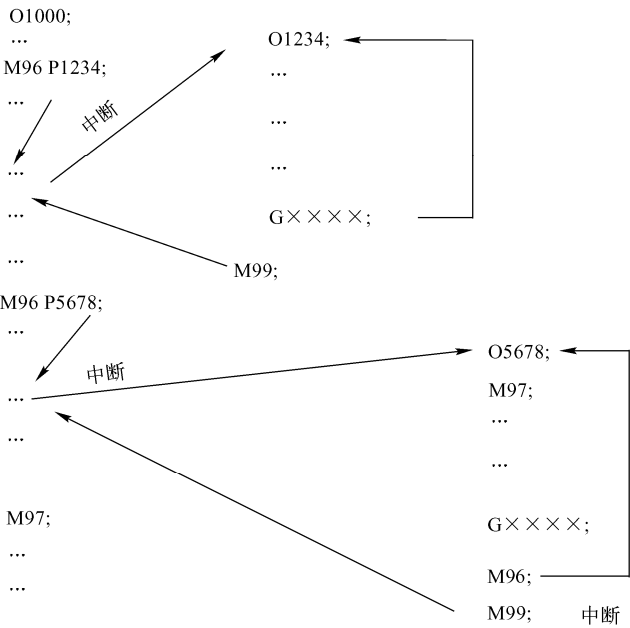
4.7.3 从用户宏程序中断返回

为把控制从用户宏程序中断返回到被中断的程序，指定 **M99**。在被中断的程

序中可以用地址 P 指定顺序号。如果这样指定，程序会从头开始检索指定的顺序号。控制返回到找到的第 1 个顺序号。

当用户宏程序中断程序正在执行时，不再产生中断。为使另一个中断有效，须执行 M99。当 M99 单独指定时，它在先前的指令结束之前执行。因此，用户宏程序中断是在中断程序中的最后指令处生效。如果这样不方便的话，应该在程序中指定 M96 和 M97，用它们控制用户宏程序的中断。

当用户宏程序中断正在执行时，没有其他用户宏程序中断产生；当产生中断时，其他中断自动地被禁止。执行 M99 使得可以产生另一个用户宏程序中断。在程序中单独指定的 M99 在前面的程序段结束之前执行。在下面的例子中，中断在 O1234 的 G××××程序段期间有效。当中断信号输入时 O1234 再次执行。O5678 由 M96 和 M97 控制。此时，O5678 中断无效（在控制返回到 O1000 之后有效）。



第 5 章 数控车床宏程序应用实例

5.1 概述

FANUC 0i-MATE-TC 系统具备宏程序功能。宏程序通过编辑其本体变量来改变刀具路径和刀具位置，特别适合图形一样，只是尺寸不同的系列零件的编程以及工艺路径一样，只是位置数据不同的系列零件的编程，还能用于抛物线、椭圆、双曲线等没有插补指令的曲线的编程。宏程序的使用不仅使得手工编程效率大大提高，更是拓宽了机床的使用范围，使数控手工编程更加灵活。

5.2 数控编程中的数学处理

编程时的数学处理，都要按零件图样、已确定的加工路线和允许的编程误差，计算出工件加工时的轨迹尺寸，即计算工件轮廓的基点或节点的坐标。对不具备刀具补偿功能的数控系统，还需计算刀具中心（刀位点）轨迹的基点和节点的坐标。

零件的轮廓一般都由许多不同的几何元素构成，如直线、圆弧、椭圆、双曲线及一般的二次曲线等，各几何元素的连接点称为基点，如相邻两直线的交点、直线与圆弧的交点或切点等。对于轮廓形状简单的零件，由于数控系统一般具有直线、圆弧插补和刀具补偿功能，数学处理只需计算零件轮廓上的基点坐标值。当零件形状复杂或零件形状与机床数控系统的插补功能不一致时，计算就比较复杂，必须用若干直线段来逼近这类曲线。要将轮廓曲线分段，用一段段直线或圆弧逼近，逼近线段与非圆曲线的交点称为节点，此时数学处理的任务是计算出各个分隔点的坐标值，并使逼近误差小于允许值。根据这些节点坐标，才能编写出轮廓曲线的加工程序。

对于没有数学方程表达式的列表曲线，如果数据点给的比较密集，则可利用这些点作为节点，用直线或圆弧连接来逼近轮廓形状。数据点比较稀疏时，通常先用插值法将节点加密或进行曲线拟合，然后再进行曲线逼近。对于空间曲面，实际是用许多的平行平面的曲线来逼近空间曲面，这时需求出所有的平面曲线，并计算各平面曲线的基点或节点，然后编写各基点、节点之间的直线或圆弧加工程序。

当采用 CAD/CAM 软件自动编程时, 这些数学处理由计算机进行。

数控机床一般都具有直线插补和圆弧插补的功能, 因此对于由直线、圆弧组成的平面轮廓零件, 它的数值计算比较简单, 主要是基点的计算。基点坐标的计算一般比较简单, 可根据零件图样给定的尺寸, 运用代数、几何、三角、解析几何的有关知识, 直接计算出数值。

数控系统一般都只有直线和圆弧插补功能, 这种机床无法直接加工除直线和圆弧以外的曲线, 如渐开线、椭圆、双曲线、阿基米德螺旋线、抛物线等。对于这些非圆曲线轮廓, 只有用直线或圆弧去逼近它, 即将轮廓曲线按编程允许的误差分割成许多小段, 再用直线或圆弧去逼近这些小段。逼近线段与非圆曲线的交点或切点称为节点, 对这种轮廓进行数学计算, 实质就是计算各节点的坐标。

1. 用直线逼近零件轮廓的节点计算

(1) 等间距法

等间距法是最简单的一种方法, 这种方法是使每个程序段的某一坐标增量相等, 然后根据曲线的表达式求出各节点的坐标值。如图 5-1 所示, 将曲线 $y=f(x)$ 的 X 轴分成等间距, 然后由起点开始, 每次增加一个坐标增量值 (间距), 根据给定的 Δx 求出 x_i , 将 x_i 代入 $y=f(x)$ 即可求出另一个坐标值 y_i , 这样依次类推, 直至求出曲线上所有相应节点的 x、y 坐标值, 并以该坐标值编制直线段程序。

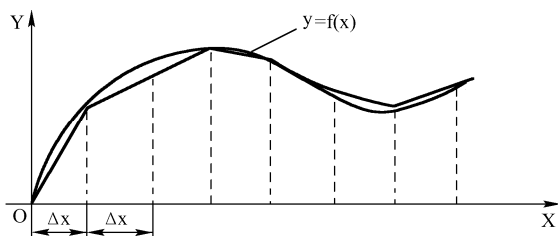


图 5-1 等间距直线逼近求节点

这种方法的关键是确定间距值, 该值应保证曲线 $y=f(x)$ 相邻两节点连线间的法向距离小于允许的程序编制误差。在实际生产中, 根据零件加工精度要求凭经验选取间距值。由于间距值 Δx 应保证曲线曲率最大处的逼近误差小于允许值, 所以程序可能过多。

(2) 等弦长法

这种方法是使所有逼近线段的弦长相等, 如图 5-2 所示。由于零件轮廓曲线的曲率各处不等, 因此各程序段的程序编制误差 δ 也不等, 这就要使各程序段的最大误差 $\delta_{\max}$ 小于允许误差 $\delta_{\text{允}}$, 才能满足程序编制的精度要求。在用直线逼近曲线时, 可以认为误差的方向是在曲线 $y=f(x)$ 的法向, 同时误差最大值发生在曲率半径最小处。

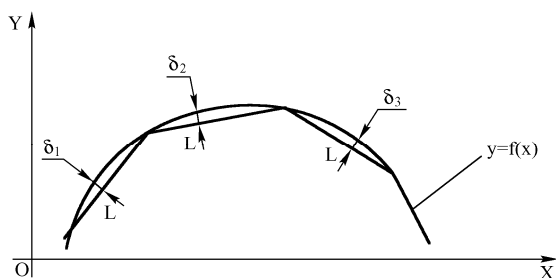


图 5-2 等弦长直线逼近求节点

等弦长法数学处理的步骤如下：

1) 确定允许的弦长 L 。由于最大误差必在曲率半径最小处，因此先确定曲率半径最小处，然后在该处按逼近误差小于或等于允许误差 $\delta_{允}$ 的条件求出允许的弦长 L ，弦长 $L = 2\sqrt{2R_{\min}\delta}$ 。

2) 用弦长分割轮廓曲线。以曲线起点 A 为圆心，作半径 L 的圆交曲线 $y=f(x)$ 于 B 点，求出 B 点的坐标。

3) 依次以 B 、 C 、 D 、 $\dots$ 为圆心，可求出其余各点的坐标值。

(3) 等误差法

等误差法是以直线拟合轮廓曲线时使所有逼近线段的误差 δ 相等，并且小于或等于允许误差，如图 5-3 所示。用这种方法确定的各逼近线段长度不等，逼近线段数目较少，但其计算过程比较复杂，要由计算机辅助完成，算法也较多，而且还在发展中。

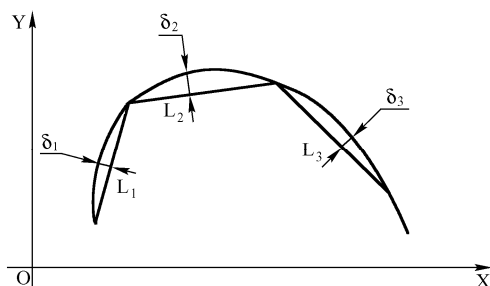


图 5-3 等误差法直线逼近求节点

2. 用圆弧逼近零件轮廓的节点计算

轮廓曲线 $y=f(x)$ 也可以用圆弧来逼近，并使逼近误差小于或等于允许误差。用圆弧逼近法去逼近零件的轮廓曲线时，需求出每段圆弧的圆心、起点和终点的坐标，以及圆弧的半径。

用圆弧逼近曲线的方法有曲率圆法、三点作圆法、相切圆法等。其中，三点作圆法、相切圆法都要先用直线逼近方法求出各节点，再通过已知节点求出圆，计算较烦琐。三点作圆法是通过已知的三个节点求圆，并作为一个圆弧插补程序段；相切圆法是通过已知的四个节点分别作出两个相切的圆，编出两个插补程序段，这种方法逼近轮廓的相邻各圆弧是相切的；曲率圆法是一种等误差圆弧逼近法。

3. 列表曲线平面轮廓的数学处理方法

列表曲线是指已经给出曲线上某些坐标,但没有给出方程,机械加工中很多轮廓曲线是用这种离散的列表点来描述的。当给出的列表曲线已密到不影响曲线的精度时,可直接在相邻列表点间用直线段或圆弧段编程,但实际上往往给出的只是很少几点,为保证精度,就要增加新的节点(也称插值)。

处理列表曲线的一般方法是采用二次拟合法,即先根据已知列表点导出插值方程(第一次曲线拟合),用数学方程式逼近列表曲线;然后再根据插值方程进行插值点加密求得新的节点(用直线段或圆弧段来逼近插值方程曲线,称二次拟合),从而编制逼近线段的程序。

5.3 圆锥曲线加工实例

圆、双曲线、抛物线、椭圆同属于圆锥曲线。早在两千多年前,古希腊数学家对它们已经很熟悉了。古希腊数学家阿波罗尼采用平面切割圆锥的方法来研究这几种曲线。用垂直于锥轴的平面去截圆锥,得到的是圆;把平面渐渐倾斜,得到椭圆;当平面和圆锥的一条母线平行时,得到抛物线;当平面再倾斜一些就可以得到双曲线。阿波罗尼曾把椭圆叫“亏曲线”,把双曲线叫做“超曲线”,把抛物线叫做“齐曲线”。

圆锥曲线的统一定义:到定点的距离与到定直线的距离比 e 是常数的点的轨迹叫做圆锥曲线。当 $0 < e < 1$ 时为椭圆;当 $e = 1$ 时为抛物线;当 $e > 1$ 时为双曲线。

5.3.1 双曲线轮廓加工

1. 定义

1) 第一定义:平面内到两定点 F_1 、 F_2 的距离之差的绝对值等于定长 $2a$ (小于 $|F_1F_2|$) 的点的轨迹叫双曲线,如图 5-4 所示。

说明:

① $||PF_1| - |PF_2|| = 2a$ ($2a < |F_1F_2|$) 是双曲线;若 $2a = |F_1F_2|$, 轨迹是以 F_1 、 F_2 为端点的射线; $2a > |F_1F_2|$ 时无轨迹。

② 设 M 是双曲线上任意一点,若 M 点在双曲线右边一支上,则 $|MF_1| > |MF_2|$, $|MF_1| - |MF_2| = 2a$;若 M 在双曲线的左支上,则 $|MF_1| < |MF_2|$, $|MF_1| - |MF_2| = -2a$,故 $|MF_1| - |MF_2| = \pm 2a$,这是与椭圆不同的地方。

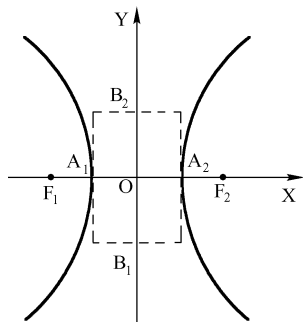


图 5-4 双曲线

2) 第二定义:平面内动点到定点 F 的距离与到定直线 L 的距离之比是常数 e

($e>1$) 的点的轨迹叫双曲线，定点叫焦点，定直线 L 叫相应的准线。

2. 范围

双曲线在不等式 $x\geq a$ 和 $x\leq -a$ 所表示的区域内。

3. 对称性

以 $-y$ 代 y ，方程不变，所以双曲线关于 X 轴对称；以 $-x$ 代 x ，方程不变，所以双曲线关于 Y 轴对称；同时以 $-y$ 代 y ，以 $-x$ 代 x ，方程不变，所以双曲线关于原点对称。

4. 顶点

令 $y=0$ ，得 $x=\pm a$ ，因此双曲线和它的一条对称轴—— X 轴有两个交点 $A_1(-a, 0)$ ， $A_2(a, 0)$ 。

令 $x=0$ 时，解得 $y^2=-b^2$ ，无实数解，说明双曲线与它的另一条对称轴—— Y 轴没有交点。

故双曲线和它的对称轴有两个交点 $A_1(-a, 0)$ 、 $A_2(a, 0)$ ，它们叫做双曲线的顶点。

线段 A_1A_2 叫双曲线的实轴，它的长等于 $2a$ ， a 叫做双曲线的实半轴长；线段 B_1B_2 叫做双曲线的虚轴，它的长等于 $2b$ ， b 叫做双曲线的虚半轴长。

5. 渐进线

1) 把两条直线 $y=\pm\frac{b}{a}x$ 叫做双曲线的渐进线；

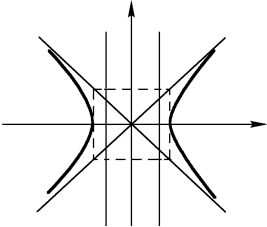
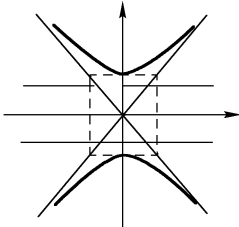
2) 从图 5-4 可以看出，双曲线 $\frac{x^2}{a^2}-\frac{y^2}{b^2}=1$ 的各支向外延伸时，与直线 $y=\pm\frac{b}{a}x$ 逐渐接近。

6. 离心率

双曲线的焦距与实轴长的比 $e=\frac{c}{a}$ ，叫双曲线的离心率， $c=\sqrt{a^2+b^2}$ 。

7. 双曲线标准方程及性质（表 5-1）

表 5-1 双曲线标准方程及性质

标准方程	$\frac{x^2}{a^2}-\frac{y^2}{b^2}=1$ ($a>0, b>0$)	$\frac{y^2}{a^2}-\frac{x^2}{b^2}=1$ ($a>0, b>0$)
图形		

(续)

标准方程	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \quad (a>0, b>0)$	$\frac{y^2}{a^2} - \frac{x^2}{b^2} = 1 \quad (a>0, b>0)$
焦点	$F_1(-c, 0), F_2(c, 0)$	$F_1(0, -c), F_2(0, c)$
顶点	$A_1(a, 0), A_2(-a, 0)$	$A_1(0, a), A_2(0, -a)$
对称轴	实轴 2a, 虚轴 2b, 实轴在 X 轴上, $c^2=a^2+b^2$	实轴 2a, 虚轴 2b, 实轴在 Y 轴上, $c^2=a^2+b^2$
离心率	$e = \frac{c}{a} = \frac{ MF_2 }{ MD }$	$e = \frac{c}{a} = \frac{ MF_2 }{ MD }$
准线方程	$l_1: x = \frac{a^2}{c}, l_2: x = -\frac{a^2}{c}$ 准线间距离为 $\frac{2a^2}{c}$	$l_1: y = \frac{a^2}{c}, l_2: y = -\frac{a^2}{c}$ 准线间距离为 $\frac{2a^2}{c}$
渐近线方程	$\frac{x}{a} + \frac{y}{b} = 0, \frac{x}{a} - \frac{y}{b} = 0$	$\frac{x}{b} + \frac{y}{a} = 0, \frac{x}{b} - \frac{y}{a} = 0$

8. 几个概念

(1) 等轴双曲线

实、虚轴相等的双曲线。等轴双曲线的渐近线为 $y = \pm x$ ，离心率为 $\sqrt{2}$ 。

(2) 共轭双曲线

以已知双曲线的实轴为虚轴，虚轴为实轴的双曲线叫原双曲线的共轭双曲线，

例： $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ 的共轭双曲线是 $\frac{x^2}{a^2} - \frac{y^2}{b^2} = -1$ 。

1) 双曲线及其共轭双曲线有共同的渐近线。但有共同渐近线的两双曲线不一定是共轭双曲线。

2) 双曲线和它的共轭双曲线的四个焦点在同一个圆周上。

9. 构建宏程序

如图 5-5 所示，编制一个车削加工带双曲线零件的程序，过渡双曲线方程为 $\frac{x^2}{a^2} - \frac{z^2}{b^2} = 1$ ，双曲线实半轴长为 a，虚半轴长为 b，使用变量编制零件加工的宏程序。

工艺分析 车削图 5-5 所示双曲线过渡的回转零件时，由于 BEIJING FANUC 0i-TC 系统只能实现工件坐标原点 Z 向偏置，故编程中首先采用直线逼近零件轮廓节点的方法，即在 Z 方向分段。为保证零件的加工精度，以 0.1 为一个步距，并把 Z 作为自变量，X 作为 Z 的函数，利用双曲线方程求出其上各点的坐

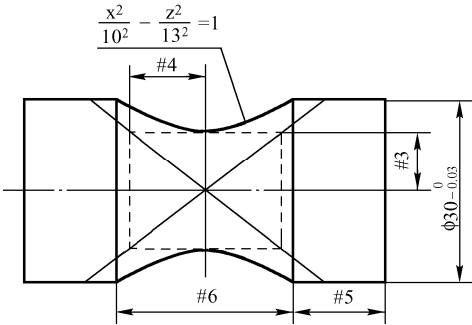


图 5-5 双曲线零件示意图

标, 然后利用数学计算使其偏移到相应位置, 并把各点连接在一起最终形成双曲线。为了适应不同的双曲线, 可以编制一个只有变量不用具体数据的宏程序, 然后在主程序中调用该宏程序的用户宏指令为上述变量赋值。这样, 对于不同的双曲线, 只需要修改主程序中用户宏指令段内的赋值数据就可以进行加工。

采用用户宏指令调用加工零件。

(1) 参数说明

#1: 双曲线任一动点 X 方向的坐标值

#2: 双曲线任一动点 Z 方向的坐标值

#3: 双曲线实半轴长

#4: 双曲线虚半轴长

#5: 双曲线起点在工件坐标系里 Z 轴上坐标值的绝对值

#6: 双曲线的开口距离

#7: X 方向的步距值

#8: 进给速度

(2) 主程序

O0001;

N05 G98 M03 S800 T0101; 零件加工工艺设置

...

N×× T0202 S1000; 调用精加工双曲线轮廓车刀及切换速度

N×× G65 P0002 C_I_J_K_D_E_; 调用精加工双曲线轮廓用户宏程序

...

N×× M05; 主轴停止

N×× M30; 程序结束并返回程序开头

(3) 用户宏程序

N×× #2=#6/2; 双曲线起点 Z 方向赋初值

N10 #1=#3*SQRT[1+[#2*#2]/[#4*#4]]; 由系统计算双曲线上任一动点的 X 坐标值

N×× G01 X[2*#1] Z[#2-#5-#6/2] F#8; 沿计算出的双曲线轨迹作直线插补

N×× #2=#2-#7; Z 方向步距递增

N×× IF [#2GE-[#6/2]] GOTO10; 循环体判断 如#2 大于等于#6/2, 则程序跳转到 N10 程序段

N×× M99; 用户宏程序结束, 返回主程序

10. 宏程序实例

编制一个车削加工带双曲线零件的程序(图 5-6), 双曲线方程为 $\frac{x^2}{a^2} - \frac{z^2}{b^2} = 1$,

双曲线的开口距离为 29.07mm，使用变量编制该零件加工的宏程序。

(1) 编程工艺设计

以工件右端面建立工件坐标系（编程坐标系与工件坐标系重合），以圆弧替代双曲线方程，利用系统提供的外圆加工循环及精加工循环对零件进行粗、精加工，然后再对双曲线轮廓进行精加工。编程中采用直线逼近零件轮廓节点的方法，即在 Z 方向分段。为保证零件的加工精度，以 0.1 为一个步距，并把 Z 作为自变量，X 作为 Z 的函数，利用双曲线方程求出其上各点的坐标，然后把各点连接在一起最终形成抛物线。

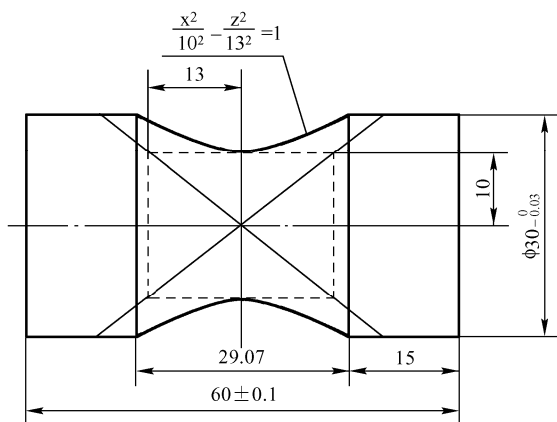


图 5-6 双曲线轮廓零件编程实例图

(2) 加工参考程序

O0001; 主程序

N05 G98 M03 S800 T0101; 零件加工工艺设置

N×× G00 X100 Z150; 刀具的起刀点

N×× M98 P0002; 调用子程序对零件进行粗、精加工

...

N×× T0202 S1000; 调用精加工双曲线轮廓车刀及切换速度

N×× G65 P0002 C10 I13 J15 K29.07 D0.1 E100; 调用精加工双曲线轮廓用户宏程序

...

N×× M05; 主轴停止

N×× M30; 程序结束并返回程序开头

子程序:

O0002; 零件轮廓粗、精加工子程序

...

...

N×× M99;

用户宏程序如下:

N×× #2=#6/2; 双曲线起点 Z 方向赋初值

N10 #1=#3*SQRT[1+[#2*#2]/[#4*#4]]; 由系统计算双曲线上任一动点的 X
坐标值

N×× G01 X[2*#1] Z[#2-#5-#6/2] F#8; 沿计算出的双曲线轨迹作直线插补

N×× #2=#2-#7; Z 方向步距递增

N×× IF [#2GE-[#6/2]] GOTO10; 循环体判断: 如果#2 大于等于#6/2, 则
程序跳转到 N10 程序段

N×× M99; 用户宏程序结束, 返回主程序

5.3.2 抛物线轮廓加工

1. 定义

平面内到一定点 F 和到一条不过此点的定直线 L 距离相等的点的轨迹叫做抛物线。定点 F 叫做抛物线的焦点, 定直线 L 叫做抛物线的准线, 如图 5-7 所示。

2. 范围

在方程 $y^2=2px$ 中, 因为 $p>0$, 所以 $x\geq 0$, 这表明除了顶点, 抛物线的图像全部落在 Y 轴的右侧。在第一象限, 随着 x 的增大, 抛物线的图像向右上方无限延伸; 在第四象限, 随着 x 的增大, 抛物线的图像向右下方无限延伸。

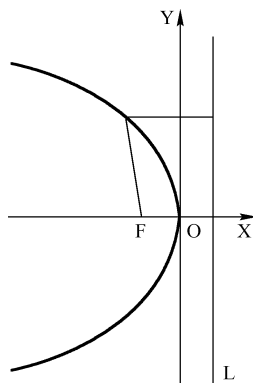


图 5-7 抛物线

3. 对称性

在方程 $y^2=2px$ 中, 以 $-y$ 换 y , 方程不变, 这表明: 如果点 $P(x, y)$ 在抛物线 $y^2=2px$ 上, 那么点 P 关于 X 轴对称的点 $P'(x, -y)$ 也在该抛物线上, 即抛物线 $y^2=2px$ 关于 X 轴对称, 是轴对称图形。

4. 顶点

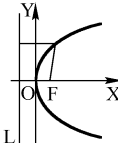
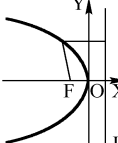
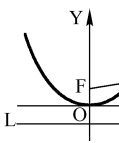
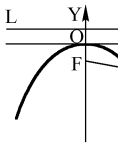
抛物线与对称轴的交点称为抛物线的顶点。抛物线 $y^2=2px$ 的顶点为坐标原点 $(0, 0)$ 。

5. 离心率

抛物线上的点 M 与焦点的距离和它到准线的距离的比, 叫做抛物线的离心率, 用 e 表示。由抛物线的定义可知, 抛物线中任一点到焦点和准线的距离都是相等的, 所以 $e=1$ 。

6. 抛物线标准方程及性质（表 5-2）

表 5-2 抛物线标准方程及性质

标准方程	$y^2=2px$ ($p>0$)	$y^2=-2px$ ($p>0$)	$x^2=2py$ ($p>0$)	$x^2=-2py$ ($p>0$)
图形				
范围	$x\geq 0$	$x\leq 0$	$y\geq 0$	$y\leq 0$
焦点	$F(\frac{p}{2}, 0)$	$F(-\frac{p}{2}, 0)$	$F(0, \frac{p}{2})$	$F(0, -\frac{p}{2})$
顶点	原点 O	原点 O	原点 O	原点 O
对称轴	关于 X 轴对称	关于 X 轴对称	关于 Y 轴对称	关于 Y 轴对称
离心率	$e=1$	$e=1$	$e=1$	$e=1$
准线方程	$x=-\frac{p}{2}$	$x=\frac{p}{2}$	$y=-\frac{p}{2}$	$y=\frac{p}{2}$

7. 构建宏程序

如图 5-8 所示，编制一个车削加工带抛物线零件的程序，抛物线方程为 $x^2=-2pz$ ，使用变量编制零件加工的宏程序。

工艺分析 车削图 5-8 所示 $x^2=-2pz$ 过渡的回转零件时，编程中首先采用直线逼近零件轮廓节点的方法，即在 X 方向分段。为保证零件的加工精度，以 0.1 为一个步距，并把 X 作为自变量，Z 作为 X 的函数，利用抛物线方程求出其上各点的坐标，然后把各点连接在一起最终形成抛物线。为了适应不同的抛物线，可以编制一个只有变量不用具体数据的宏程序，然后在主程序中调用该宏程序的用户宏指令为上述变量赋值。这样只需要修改主程序中用户宏指令段内的赋值数据就可以直接进行加工。

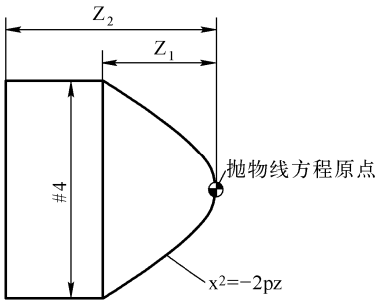


图 5-8 抛物线曲线零件示意图

采用用户宏指令调用加工零件。

(1) 参数说明

#1：抛物线任一动点 X 方向的坐标值

#2：抛物线任一动点 Z 方向的坐标值

#3; 抛物线焦点坐标值在 Z 轴上坐标值的绝对值

#4; 抛物线的开口距离

#5; X 方向的步距值

#6; 进给速度

(2) 主程序

O0001;

N05 G98 M03 S800 T0101; 零件加工工艺设置

...

N×× T0202 S1000; 调用精加工抛物线轮廓车刀及切换速度

N×× G65 P0002 A_B_C_I_J_K_; 调用精加工抛物线轮廓用户宏程序

...

N×× M05; 主轴停止

N×× M30; 程序结束并返回程序开头

(3) 用户宏程序

O0002;

N05 G00 X#1 Z[#2+5]; 刀具快速接近抛物线顶点处定位

N×× G01 Z#2 F#6; 以工件进给速度直线插补到抛物线顶点处

N10 #1=#1+#5; X 方向步距递增

N×× #2=-[#1*#1]/[2*#3]; 由系统计算抛物线上任一动点的 Z 坐标值

N×× G01 X[2*#1] Z#2 F#6; 沿计算出的抛物线轨迹作直线插补

N×× IF [#1LE[#4/2]] GOTO10; 循环体判断 如果#1 小于#4/2, 则程序跳转到 N10 程序段

N×× G00 X[#4+2]; 循环体结束后刀具离开已加工表面

N×× Z5; 刀具退回工件右端面外

N×× M99; 用户宏程序结束, 返回主程序

8. 宏程序实例

编制一个车削加工带抛物线零件的程序, 如图 5-9 所示。抛物线方程为 $x^2 = -16z$, 抛物线的开口距离为 32mm, 使用变量编制该零件加工的宏程序。

(1) 编程工艺设计

以工件右端面建立工件坐标系 (编程坐标系与工件坐标系重合), 以圆弧替代抛物线方程, 利用系统提供的外圆加

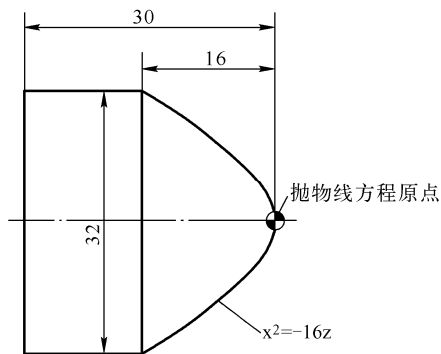


图 5-9 抛物线轮廓零件编程实例图

工循环及精加工循环对零件进行粗、精加工，然后再对抛物线轮廓进行精加工。编程中采用直线逼近零件轮廓节点的方法，即在 X 方向分段。为保证零件的加工精度，以 0.1 为一个步距，并把 X 作为自变量，Z 作为 X 的函数，利用抛物线方程求出其上各点的坐标，然后把各点连接在一起最终形成抛物线。

(2) 加工参考程序

O0001; 主程序

N05 G98 M03 S800 T0101; 零件加工工艺设置

N×× G00 X100 Z150; 刀具的起刀点

N×× M98 P0002; 调用子程序对零件进行粗、精加工

...

...

N×× T0202 S1000; 调用精加工抛物线轮廓车刀及切换速度

N×× G65 P0003 A0 B0 C8 I32 J0.1 K100; 调用精加工抛物线轮廓用户宏程序

...

...

N×× M05; 主轴停止

N×× M30; 程序结束并返回程序开头

子程序:

O0002; 零件轮廓粗、精加工子程序

...

...

N×× M99

用户宏程序:

O0003;

N05 G00 X#1 Z[#2+5]; 刀具快速接近抛物线顶点处定位

N×× G01 Z#2 F#6; 以工件进给速度直线插补到抛物线顶点处

N10 #1=#1+#5; X 方向步距递增

N×× #2=-[#1*#1]/[2*#3]; 由系统计算抛物线上任一动点的 Z 坐标值

N×× G01 X[2*#1] Z#2 F#6; 沿计算出的抛物线轨迹作直线插补

N×× IF [#1LE[#4/2]] GOTO10; 循环体判断 如果#1 小于#4/2, 则程序跳转到 N10 程序段

N×× G00 X[#4+2]; 循环体结束后刀具离开已加工表面

N×× Z5; 刀具退回工件右端面外

N×× M99; 用户宏程序结束, 返回主程序

5.3.3 椭圆轮廓加工

1. 定义

椭圆是平面上到两定点的距离之和为常值的点的轨迹，也可定义为到定点距离与到定直线间距离之比为常值的点的轨迹，如图 5-10 所示。

2. 椭圆方程

1) 椭圆的标准方程 $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$,

其中 $b^2 = a^2 - c^2$, 满足 $a > b > 0$ 。

2) 椭圆的参数方程:

$$\begin{cases} x = a \cos \varphi \\ y = b \sin \varphi \end{cases} \text{ 其中 } a > b > 0, \varphi \text{ 是参数。}$$

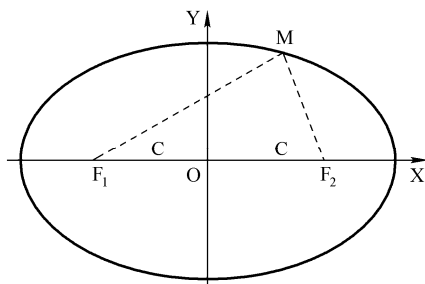


图 5-10 椭圆

3. 范围

由椭圆的标准方程知，椭圆上点的坐标 (x, y) 都适合不等式 $x^2/a^2 \leq 1$, $y^2/b^2 \leq 1$, 说明椭圆位于直线 $x = \pm a$ 和直线 $y = \pm b$ 所围成的矩形里。

4. 对称性

在椭圆的标准方程中，把 x 换成 $-x$ ，或把 y 换成 $-y$ ，或把 x 、 y 同时换成 $-x$ 、 $-y$ 时，方程都不变，所以图形关于 Y 轴、 X 轴或原点对称。这时，坐标轴是椭圆的对称轴，原点是椭圆的对称中心。

5. 顶点

在椭圆的标准方程中，令 $x=0$ ，得 $y=\pm b$ ，点 $B_1(0, -b)$ 、 $B_2(0, b)$ 是椭圆和 Y 轴的两个交点；令 $y=0$ ，得 $x=\pm a$ ，点 $A_1(-a, 0)$ 、 $A_2(a, 0)$ 是椭圆和 X 轴的两个交点。 Y 轴、 X 轴是椭圆的对称轴，椭圆与 Y 轴、 X 轴有四个顶点 $A_1(-a, 0)$ 、 $A_2(a, 0)$ 、 $B_1(0, -b)$ 、 $B_2(0, b)$ ，这四个点即是椭圆的顶点。

线段 A_1A_2 、 B_1B_2 分别叫做椭圆的长轴、短轴。它们的长分别等于 $2a$ 和 $2b$ ， a 和 b 分别叫做椭圆的长半轴长和短半轴长。

6. 离心率

椭圆焦距与长轴长之比 $e=c/a$ ，叫做椭圆的离心率。

因为 $a > b > 0$ ，所以 $0 < e < 1$ 。 e 越接近 1，则 c 越接近 a ，从而 $b = \sqrt{a^2 - c^2}$ 越小，因而椭圆越扁；反之， e 越接近 0，则 b 越接近 a ，这时椭圆就接近于圆。

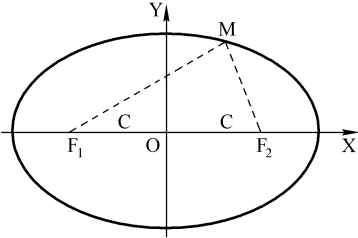
如果 $a=b$ ，则 $c=0$ ，两个焦点重合，这时椭圆的标准方程为

$$x^2 + y^2 = a^2$$

椭圆就变成圆。

7. 椭圆标准方程及性质（表 5-3）

表 5-3 椭圆标准方程及性质

椭圆标准方程	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (a > b > 0)$
a、b、c 的关系	$c^2 = a^2 - b^2 \quad (a > b > 0)$
图形	
范围	$ x \leq a, y \leq b$
对称性	椭圆是轴对称图形（对称轴是 X 轴、Y 轴）；椭圆也是中心对称图形（对称中心是原点）
顶点	$A_1(-a, 0)$ 、 $A_2(a, 0)$ 、 $B_1(0, -b)$ 、 $B_2(0, b)$ 长轴长 2a，短轴长 2b
离心率	$e = \frac{c}{a}$ ， $(0 < e < 1)$ ，e 越接近于 1，椭圆越“扁”，反之，椭圆越“圆”
渐近线	无

8. 构建宏程序

如图 5-11 所示，编制一个车削加工右 1/2 椭圆零件的回转轮廓线程序，椭圆方程为 $x^2/a^2 + z^2/b^2 = 1$ ，使用变量编制零件加工的宏程序。

工艺分析 以单层宏程序为例，欲车削图 5-11 所示 $x^2/a^2 + z^2/b^2 = 1$ 椭圆的回转零件时，在数控程序中用任意一点 D 的 z 值（用 2 号变量）来表达该点的 x 值（用 5 号变量）；根据椭圆方程： $x^2/a^2 + z^2/b^2 = 1$ 得

$x = a\sqrt{1 - z^2/b^2}$ ，加工从 A（x=0，z=b）点到 B（x=a，z=0）点，z 值从 b 到 0 以一定值递减，相应地 x 值从 0 到 a 按计算公式递增。编程中首先采用直线逼近零件轮廓节点的方法，即在 Z 方向分段。为保证零件的加工精度，以 0.1 为一个步距，并把 Z 作为自变量，X 作为 Z 的函数，利用椭圆方程求出其上各点的坐标，

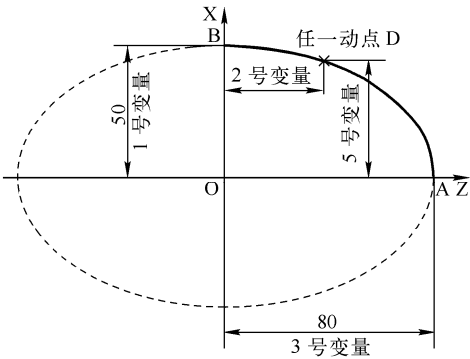


图 5-11 椭圆曲线零件示意图

然后把各点连接在一起最终形成椭圆。为了适应不同的椭圆，可以编制一个只有变量不用具体数据的宏程序，然后在主程序中调用该宏程序的用户宏指令为上述变量赋值。这样只需要修改主程序中用户宏指令段内的赋值数据就可以直接进行加工。

采用用户宏指令调用加工零件。

(1) 参数说明

#1; (A) 椭圆短轴半径

#2; (B) 椭圆上任意一点 Z 坐标值

#3; (C) 椭圆长轴半径

#5; (J) 椭圆上任意一点 X 坐标值

#6; (K) Z 坐标值每次变化量 1

#7; (D) 0

#8; 进给速度

(2) 主程序

O0001;

N5...

...

N×× G65 P0002 A50 B80 C80 DO K1 E100; 调用椭圆轮廓用户宏程序

...

N×× M30; 程序结束并返回程序开头

(3) 用户宏程序

O0002;

N5 #5=[#1+#1]*SQRT[1-#2*#2/#3/#3]; 由系统计算椭圆上任一动点的 X 坐标值

N×× G1 X#5 Z#2 #8; 沿计算出的椭圆轨迹作直线插补

N×× #2=#2-#6; Z 方向步距递增

N×× IF [#2 GE #7] GOTO 5; 循环体判断 如果#2 小于#7, 则程序跳转到 N5 程序段

N×× M99; 用户宏程序结束, 返回主程序

9. 宏程序实例

加工图 5-12 中椭圆部分, 编制一个车削加工带椭圆零件的程序, 椭圆方程为 $x^2/22^2 + z^2/25^2 = 1$, 使用变量编制该零件加工的宏程序。

(1) 编程工艺设计

以工件右端面建立工件坐标系 (编程坐标系与工件坐标系重合), 以椭圆的直径取适当倍数为毛坯值, 根据工件材料、刀具等要素确定背吃刀量, 椭圆短轴直

径为比较值，建立分层循环（图 5-13）。以椭圆任一动点的 Z 坐标为自变量，X 为应变变量，利用椭圆公式完成单层椭圆的加工。加工过程中，以前一刀加工时的短轴直径减背吃刀量与椭圆的实际短轴直径相比较构成分层循环，大于、等于椭圆的实际短轴直径时分层切削。由于先运算后比较存在欠切现象，则判断小于椭圆的实际短轴直径减背吃刀量时，以椭圆的实际短轴直径赋值，完成椭圆加工，避免欠切现象。

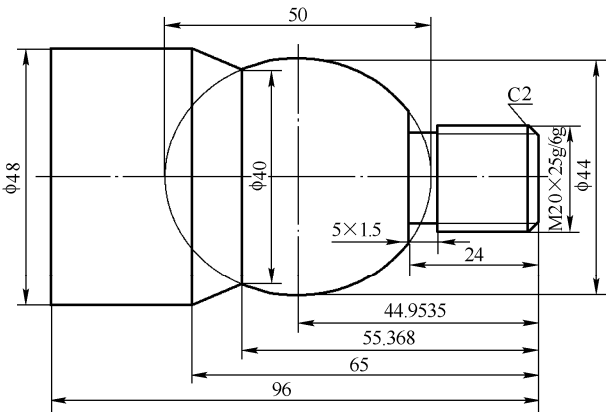


图 5-12 椭圆轮廓零件编程实例图

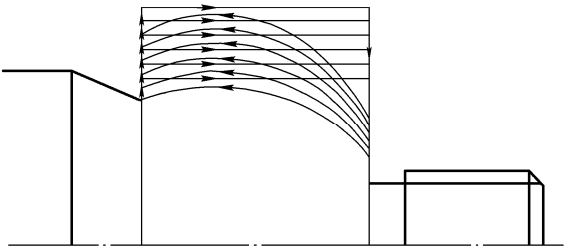


图 5-13 椭圆轮廓加工刀具轨迹图

(2) 加工参考程序

1) 参数说明：

- #1；椭圆任一动点的 X 直径值
- #2；椭圆任一动点的 Z 值，动点距椭圆坐标系原点 Z 向距离
- #3；椭圆加工时短轴赋值
- #4；椭圆长半轴
- #5；椭圆坐标系原点与工件坐标系原点的距离
- #6；步距（自变量 Z 轴方向上的增量）
- #7；椭圆任一动点 Z 向的终点，动点距椭圆坐标系原点 Z 向距离

#8; X 方向上的背吃刀量

#9; 椭圆短轴直径

#10; 椭圆 Z 向起刀点与工件坐标系原点的距离

2) 主程序:

O9501;

N×× G98 M03 S500 T0101 F150; 零件加工工艺设置

N×× G00 X100 Z100; 定义刀具的起刀点

N×× ...

N×× G65 P9502 I25 J-44.9535 K0.1 I-55.368 J2 K44 I-24; 调用椭圆轮廓用户宏程序

N×× ...

N×× G00 X100; 返回刀具的起刀点

N×× Z150;

N×× M05; 主轴停止

N×× M30; 程序结束并返回程序开头

3) 用户宏程序:

O9502;

N×× #3=3*[#9/2]; 以椭圆短轴为参考确立刀具 X 向初始值

N×× G00 Z5; 刀具接近工件

N×× Z#10; 刀具运动到椭圆 Z 向起刀点

N×× X[#9+4]; 刀具运动到椭圆 X 向起刀点

N05 #2=#10-#5; 计算椭圆任一动点 Z 向的初始值, 动点距椭圆坐标系原点 Z 向距离

N10 #1=#3*SQRT[1-[#2*#2]/[#4*#4]]; 计算椭圆任一动点 X 向的初始值

N×× G01 X#1 Z[#2+#5]; 沿计算出的椭圆轨迹作直线插补

N×× #2=#2-#6; Z 方向步距递增

N×× IF [#2GE[#7-#5]] GOTO10; 循环体判断 如果#2 大于等于#7-#5, 则程序跳转到 N10 程序段

N×× G00 X[#3+2]; 刀具离开已加工表面

N×× Z#10; 刀具运动到椭圆 Z 向起刀点

N×× #3=#3-#8; 椭圆短轴赋值计算

N×× IF [#3GE#9] GOTO05; 循环体判断 如果#3 大于等于#9, 则程序跳转到 N05 程序段

N×× IF [#3GT[#9-#8]] GOTO15; 循环体判断 如果#3 大于#9-#8, 则程序跳转到 N15 程序段


```
N×× IF [#3GE[#9-#8]] GOTO20; 循环体判断 如果#3 等于#9-#8, 则程序
                                跳转到 N20 程序段
N×× N15#3=#9; #3 赋椭圆短轴直径值
N×× #2=#10-#5; 计算椭圆任一动点 Z 向的初始值, 动点距椭圆坐标系原
                                点 Z 向距离
N25 #1=#3*SQRT[1-[#2*#2]/[#4*#4]]; 计算椭圆任一动点 X 向的初始值
N×× G01 X#1 Z[#2+#5]; 沿计算出的椭圆轨迹作直线插补
N×× #2=#2-#6; Z 方向步距递增
N×× IF [#2GE[#7-#5]] GOTO25; 循环体判断 如果#2 大于等于#7-#5, 则
                                程序跳转到 N25 程序段
N×× G00 X[#3+2]; 刀具离开已加工表面
N20 G00 X100; 刀具返回起刀点
N×× Z150;
N×× M99; 用户宏程序结束, 返回主程序
```

5.4 螺纹加工实例

5.4.1 梯形螺纹的基本知识

1. 梯形外螺纹基本尺寸的计算

梯形螺纹分米制和寸制两种, 我国常用牙型角为 30°的米制梯形螺纹, 而牙型角为 29°的寸制梯形螺纹在我国则较少应用。

要正确车削梯形外螺纹, 首先要掌握它的基本结构和相关参数。图 5-14 给出了梯形螺纹的牙型图, 其基本要素的计算公式及计算实例见表 5-4。若为标准梯形螺纹, 则其基本尺寸和参数可通过查表获得。

表 5-4 梯形螺纹基本要素计算公式 (单位: mm)

名 称		计 算 公 式			
牙型角 α		$\alpha=\alpha'=30^\circ$			
螺距 t		由螺纹标准确定			
		t	2~4	5~12	16~48
间隙 Z		Z	0.25	0.5	1
外螺纹	大径 d	公称直径			
	中径 d_2	$d_2=d-0.5t$			
	小径 d_3	$d_3=d-2h_3$			
	牙型高度 h_3	$h_3=0.5t+Z$			

(续)

名 称		计 算 公 式		
内 螺 纹	大径 D_4	$D_4=d+2Z$		
	中径 D_2	$D_2=d_2$		
	小径 D_1	$D_1=d-t$		
	牙型高度 h_4	$h_4=h_3$		
牙顶宽 f	$f=f'=0.366t$			
牙槽底宽 W	t	2~4	5~12	16~48
	$W'=W$	0.366t-0.13	0.366t-0.27	0.366t-0.54
螺旋角 λ	$\tan \lambda = \frac{\text{tn}}{\pi d_2}$			

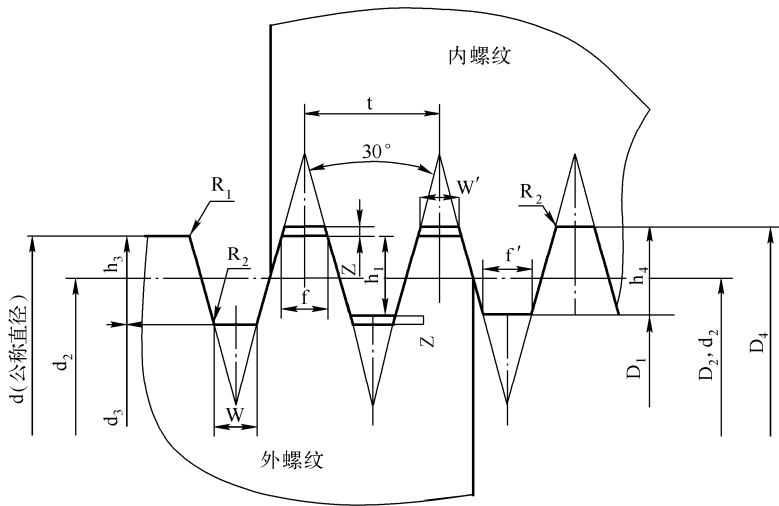


图 5-14 梯形螺纹牙型图

2. 梯形螺纹的一般技术要求

梯形螺纹的一般技术要求见表 5-5。

表 5-5 梯形螺纹的一般技术要求

项 目	具 体 内 容
1	梯形螺纹的中径必须与基准轴颈同轴，其大径尺寸应小于基本尺寸
2	梯形螺纹的配合以中径定心，因此车削梯形螺纹必须保证中径尺寸公差
3	梯形螺纹牙型角要正确
4	梯形螺纹牙型两侧面的表面粗糙度值要小

3. 梯形螺纹加工用车刀

(1) 梯形螺纹车刀的选择

梯形螺纹车刀一般分为高速钢车刀和硬质合金车刀两大类。低速车削时一般选用高速钢车刀，而加工一般精度的梯形螺纹时可采用硬质合金车刀进行高速车削。由于梯形螺纹的牙型较深，车削时切削抗力较大，粗车时常采用弹性刀排。

(2) 梯形螺纹车刀的几何角度

1) 高速钢梯形外螺纹车刀的几何角度。车削梯形外螺纹时，为减小切削力，螺纹车刀分粗车刀和精车刀两种，如图 5-15 所示。

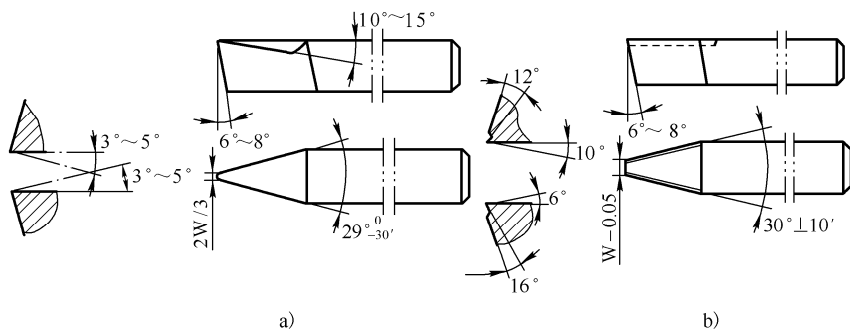


图 5-15 高速钢梯形外螺纹车刀

a) 粗车刀 b) 精车刀

2) 硬质合金梯形外螺纹车刀的几何角度。硬质合金梯形外螺纹车刀及其几何角度如图 5-16 所示。加工时，由于三个切削刃同时参与切削，切削力较大，容易引起振动。

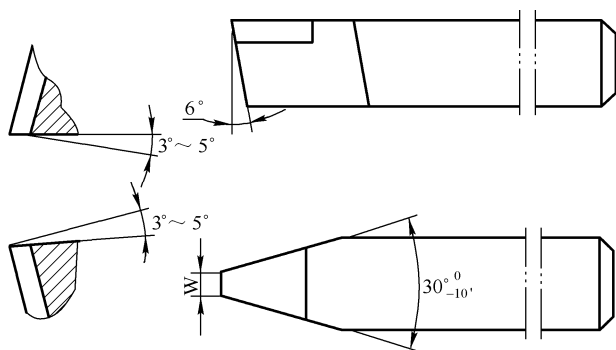


图 5-16 硬质合金梯形外螺纹车刀

3) 梯形内螺纹车刀的几何角度。梯形内螺纹车刀与内管螺纹车刀基本相同，只是刀尖角等于 30° ，如图 5-17 所示。

需要指出的是，为了增加刀头强度、减小振动，梯形内螺纹车刀的前面应适当磨低一些。

(3) 梯形外螺纹车刀的几何角度选取

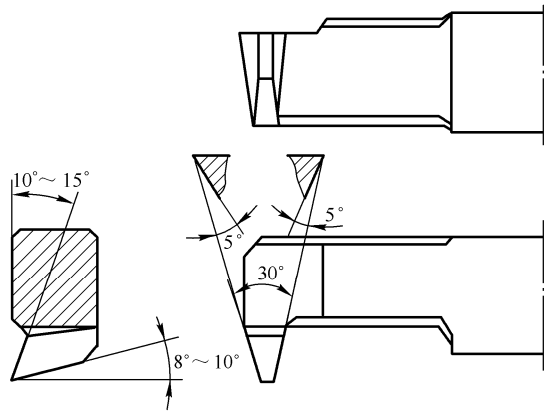


图 5-17 梯形内螺纹车刀

梯形外螺纹车刀的几何角度选取见表 5-6，梯形内螺纹车刀的几何角度选取与此类似，可参照选取。

表 5-6 梯形外螺纹车刀的几何角度选取

类型	参 数 选 取
粗车刀	1) 刀尖角略小于梯形螺纹牙型角，一般取 29° 2) 刀尖宽度小于牙型槽底宽 W，一般取 2W/3 3) 径向前角取 10°~15°，径向后角取 6°~8° 4) 两侧后角进刀方向取 (3°~5°)+ψ，背进刀方向取 (3°~5°)-ψ 5) 刀尖处适当倒圆
精车刀	1) 刀尖角等于梯形螺纹牙型角，即 0° 2) 刀尖宽度等于牙型槽底宽 W 减去 0.05mm 3) 径向前角取 0°，径向后角取 6°~8° 4) 两侧后角进刀方向取 (5°~8°)+ψ，背进刀方向取 (5°~8°)-ψ 5) 两侧磨有前角 10°~20°的卷屑槽

4. 梯形螺纹车刀的刃磨

(1) 梯形螺纹车刀的刃磨要求（表 5-7）

表 5-7 梯形螺纹车刀的刃磨要求

序号	内 容
1	刃磨两刃夹角时，应随时目测和用样板校对
2	径向前角不为零时，两刃的夹角应修正，其修正方法同管螺纹车刀修正方法
3	各切削刃应光滑、平直、无裂口，两侧切削刃应对称，刀体不歪斜
4	各切削刃用油石去毛刺

梯形螺纹车刀刃磨的主要参数：螺纹的牙型角和牙底槽宽度

(2) 梯形螺纹车刀刃磨的注意事项（表 5-8）

表 5-8 梯形螺纹车刀刃磨的注意事项

序号	内 容
1	刃磨两侧后角时，要注意螺纹的左右旋向，并根据螺纹升角口的大小来确定两侧后角的增减
2	刃磨高速钢梯形螺纹车刀时，应随时蘸水冷却，以防刃口退火
3	螺距较小的梯形螺纹精车刀不便于刃磨断屑槽时，可采用较小径向前角的梯形螺纹精车刀

(3) 梯形螺纹车刀的刃磨过程

- 1) 操作准备。准备好高速钢刀具材料、刀具图样、细粒度砂轮（如 80#白刚玉砂轮）、防护眼镜、冷却用水、油石、角度尺和样板。
- 2) 操作过程。梯形外螺纹车刀的刃磨。

5. 梯形螺纹车刀的安装

装夹梯形螺纹车刀时，车刀的主切削刃必须与工件轴线等高（用弹性刀杆应高于轴线约 0.2mm），同时应和工件轴心线平行。刀头的角平分线要垂直于工件轴线。如图 5-18 所示，用样板找正装夹，以免产生螺纹半角误差。

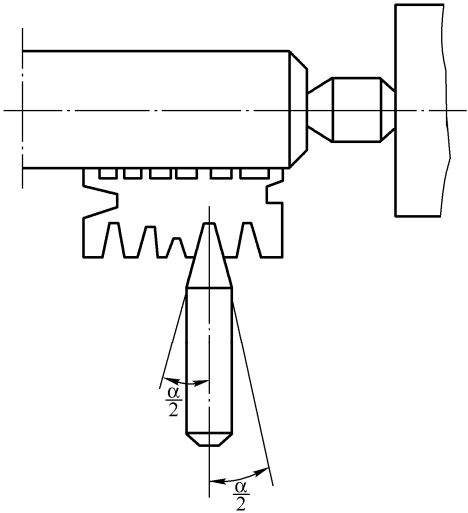


图 5-18 梯形螺纹车刀的安装

6. 梯形螺纹的加工

(1) 梯形外螺纹的车削方法

在车削梯形外螺纹时，首先看工件的螺距大小和精度要求，然后决定它的车削方法。

对于螺距小于 4mm、精度要求不高的梯形外螺纹，可用一把梯形外螺纹车刀

车削完成。粗车时可采用左右切削法，精车时采用斜进法，如图 5-19 所示。

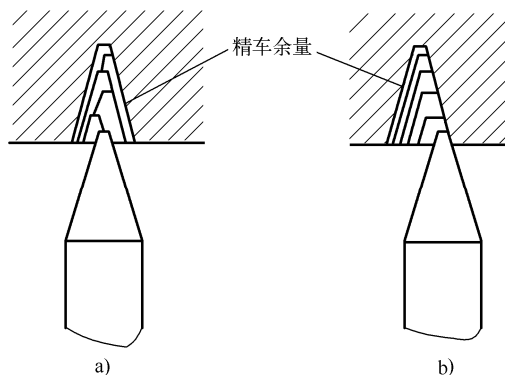


图 5-19 螺距小于 4mm 的车削方法

a) 左右切削法 b) 斜进法

螺距大于 4mm 或者精度要求较高的梯形外螺纹，一般采用分刀车削的方法，即用两把或三把刀车削完成，如图 5-20、图 5-21 所示。加工操作详细步骤见表 5-9。

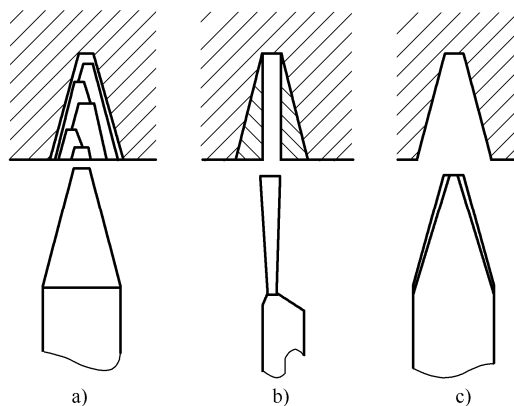


图 5-20 螺距为 4~8mm 的车削方法

a) 左右切削法粗、精车 b) 车直槽法粗车 c) 左右切削法精车

(2) 梯形内螺纹的车削

梯形内螺纹的车削方法与内管螺纹的车削方法基本相同，主要步骤如下。

- 1) 加工内螺纹底孔， $D_{\text{孔}} = D_1 = d - P$ 。
- 2) 在端面上车一个轴向深度为 1~2mm、孔径等于螺纹基本尺寸的内台阶孔，作为车内螺纹时的对刀基准。
- 3) 粗车内螺纹，采用斜进法（向背进刀方向赶刀，以利于粗车的顺利进行）。车刀刀尖与对刀基准间保证有 0.10~0.15mm 的间隙。

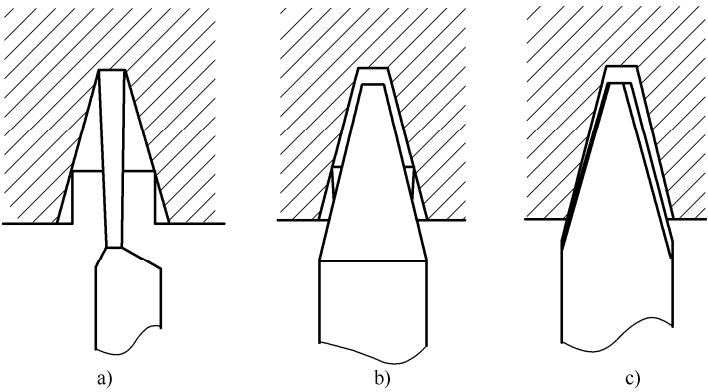


图 5-21 螺距大于 8mm 的车削方法

a) 车阶梯槽 b) 左右切削法半精车两侧面 c) 左右切削法精车

表 5-9 梯形外螺纹的分刀车削

类型	步 骤
螺距为 4~8mm 或精度要求高	粗车、半精车螺纹大径，留精车余量 0.3mm 左右，倒角（与端面成 15°）
	用左右切削法粗、半精车螺纹，每边留精车余量 0.1~0.2mm，螺纹小径精车至尺寸。或选用刀头宽度稍小于槽底宽的切断刀，采用直进法粗车螺纹，槽底直径等于螺纹小径
	精车螺纹大径至图样要求
	用两侧切削刃磨有卷屑槽的梯形螺纹精车刀，采用左右切削法精车两侧面至图样要求
螺距大于 8mm	粗车、半精车螺纹大径，留 0.3mm 左右精车余量，倒角（与端面成 15°）
	用刀头宽度小于 P/2 的切槽刀直进法粗车螺纹至接近中径处，再用刀头宽度小于槽底宽的切槽刀直进法粗车螺纹，槽底直径等于螺纹小径，形成阶梯状的螺旋槽
	用梯形螺纹粗车刀，采用左右切削法半精车螺纹槽两侧面，每面留精车余量 0.1~0.2mm
	精车螺纹大径至图样要求
	用梯形螺纹精车刀，采用左右切削法精车两侧面，控制中径，完成螺纹加工

4) 精车内螺纹，采用左右切削法精车牙型两侧面。车刀刀尖与对刀基准相接触。

由于梯形内螺纹车刀与内管螺纹车刀基本相同，只是刀尖角等于 30°，这里不再赘述。需要提醒的是，为了增加刀头强度、减小振动，梯形内螺纹车刀的前面应适当磨低一些。

7. 指令介绍

(1) 单行程螺纹切削指令 G32

FANUC 0i-MATE-TC 系统格式：G32 X (U) __Z (W) __F__

X、Z：取值为螺纹终点坐标值；

U、W：取值为螺纹终点相对循环起点的坐标分量；

F: 螺纹的导程。

(2) 螺纹切削循环指令 G92

格式: G92 X (U) Z (W) R F

(3) 螺纹切削复合循环指令 G76

G76 螺纹切削复合循环指令较 G92 指令简捷, 可节省程序设计与计算时间, 只需指定一次有关参数, 则螺纹加工过程自动进行。如图 5-22 所示为复合螺纹切削循环的刀具加工路线。G76 螺纹切削指令的格式需要同时用两条指令来定义。

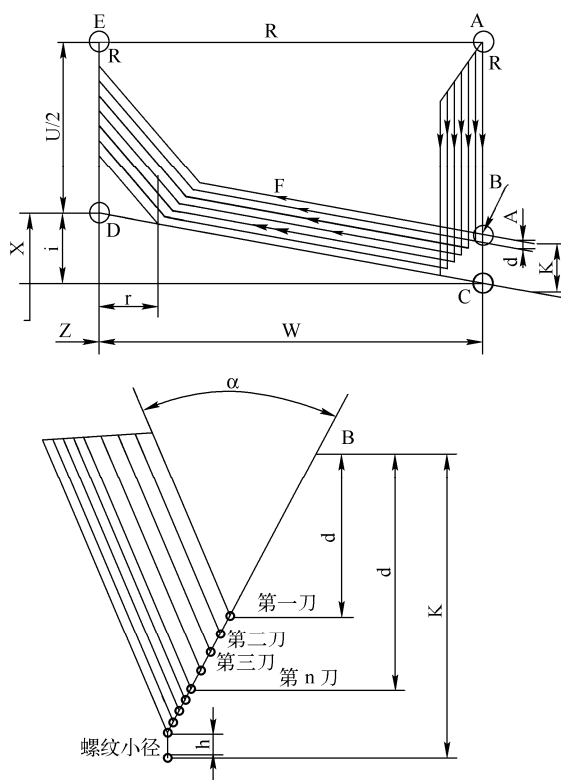


图 5-22 FANUC 0i-MATE-TC 螺纹切削复合循环轨迹

编程格式: G76 P (m) (r) (α) Q ($\Delta d_{\min}$) R (d);

G76 X (U) Z (W) R (i) P (k) Q (Δd) F (L)

式中 m、r、 α ——m 为精车重复次数, 从 01 到 99, 用两位数表示, 该参数为模态量; r 为螺纹尾端倒角值, 该值的大小可设置在 0.0~9.9L 之间, 系数应为 0.1 的整倍数, 用 00~99 之间的两位整数来表示, 其中 L 为导程, 该参数为模态量; α 为刀尖角度, 可从 80°、60°、55°、30°、29°、0° 六个角度中选择, 用两位整数来

表示,该参数为模态量;用地址 P 同时指定,例如, $m=2, r=1.2L, \alpha=60^\circ$, 表示为 P021260;

$\Delta d_{\min}$ ——最小车削深度,用半径编程指定,单位为微米;车削过程中每次的车削深度为 $\Delta d_{\min}$,当计算深度小于此极限值时,车削深度锁定在这个值,该参数为模态量;

d——精车余量,用半径编程指定,单位为毫米,该参数为模态量;

X (U)、Z (W)——螺纹终点绝对坐标或增量坐标;

i——螺纹锥度值,用半径编程指定,如果 $i=0$ 则为直螺纹,可省略;

k——螺纹高度,用半径编程指定,单位为微米;

Δd ——第一次车削深度,用半径编程指定,单位为微米;

L——螺纹导程。

(4) 使用螺纹复合循环指令时的注意事项

1) G76 可以在 MDI 方式下使用。

2) 在执行 G76 循环时,如果按下循环暂停键,则刀具在螺纹切削后的程序段暂停。

3) G76 指令为非模态指令,所以必须每次指定。

4) 在执行 G76 时,如要进行手动操作,刀具应返回到循环操作停止的位置。如果没有返回到循环停止位置就重新启动循环操作,手动操作的位移将叠加在该条程序段停止时的位置上,刀具轨迹就多移动了一个手动操作的位移量。

(5) 加工梯形螺纹时 Z 向刀具偏置值的计算

在梯形螺纹的实际加工中,由于刀尖宽度并不等于槽底宽,在经过一次 G76 切削循环后,仍无法正确控制螺纹中径等各项尺寸。为此,可经刀具 Z 向偏置后,再次进行 G76 循环加工,即可解决以上问题。为了提高加工效率,最好只进行一次偏置加工,故必须精确计算 Z 向的偏置量。

实际加工时,在一次循环结束后,用三针测量实测 M 值,计算出刀具 Z 向偏置量,然后在刀长补偿或磨耗存储器中设置 Z 向刀偏量,再次用 G76 循环加工就能一次性精确控制中径等螺纹参数值。

8. 梯形螺纹的测量

梯形螺纹的测量分三针测量、单针测量和综合测量三种。梯形外螺纹的主要测量参数包括:螺距、大径和中径。

(1) 三针测量法

三针测量法是一种比较精密的检测方法,适用于测量精度要求较高、螺纹升角小于 4° 的螺纹中径。三针测量时,将 3 根直径相等、尺寸合适的量针放置在梯形螺纹两侧对应的螺旋槽中,用千分尺测量两边量针顶点之间的距离 M,由 M 值换算出梯形螺纹中径 d_1 的实际尺寸。量针直径 d_D 不能太大,必须保证量针截面与

梯形螺纹牙侧相切；也不能太小，否则量针将陷入牙槽中，其顶点低于梯形螺纹牙顶而无法测量。选用量针时，应尽量接近最佳值，以便获得较高的测量精度，有关计算公式见表 5-10。

表 5-10 M 值及量针直径的计算公式

螺纹或蜗杆	牙型角	M 值计算公式	量针直径 d_D		
			最大值	最佳值	最小值
梯形螺纹	30°	$M=d_2+4.864d_D-1.866P$	0.656P	0.518P	0.486P
米制蜗杆	40°	$M=d_1+3.924d_D-4.316m_x$	2.446 m_x	1.672 m_x	1.61 m_x
普通螺纹	60°	$M=d_2+3d_D-0.866P$	1.01P	0.577P	0.505P
寸制螺纹	55°	$M=d_2+3.166d_D-0.961P$	0.894P-0.029	0.564P	0.481P-0.016

当采用三针测量 Tr36×6-7h 梯形螺纹中径时，量针直径的最佳值为 $d=0.518P=0.518\times 6\text{mm}=3.108\text{mm}$ ，现选用量针直径 $d_D=3.177\text{mm}$ ；螺纹直径 $d_2=d-0.5P=(36-0.5\times 6)\text{mm}=33\text{mm}$ ；由梯形螺纹公差标准查得中径尺寸和上、下偏差；对于 M 值， $M=d_2+4.864d_D-1.866P=(33+4.864\times 3.177-1.866\times 6)\text{mm}=37.257\text{mm}$ 。根据中径允许的极限偏差，千分尺的读数 M 值应为 36.922~37.257mm。

(2) 单针测量法

在测量直径和螺距较大的螺纹中径时，用单针测量比用三针测量方便、简单。测量时将一根量针放入螺旋槽中，另一侧则以螺纹的大径为基准，用千分尺测量出量针顶点与另一侧螺纹大径之间的距离 A，由 A 值换算出螺纹中径的实际尺寸。量针的选择与三针测量相同。

在单针测量前，应先量出螺纹大径的实际尺寸 d_0 ，并根据选用量针的直径 d_D 计算出用三针测量时的 M 值，然后计算出 A 值， $A=(M+d_0)/2$ 。

(3) 综合测量法

对于精度要求不高的梯形外螺纹，一般采用标准的梯形螺纹量规——螺纹环规进行综合检测。检测前，应先检查螺纹的大径、牙型角和牙型半角、螺距和表面粗糙度，然后用螺纹环规检测。如果螺纹环规的通规能顺利拧入工件螺纹，而止规不能拧入，则说明被检梯形螺纹合格。

9. 梯形螺纹的质量分析

梯形螺纹加工质量问题与普通管螺纹的质量问题类似，主要有以下几个方面。

- 1) 中径不正确。主要由于车刀背吃刀量不正确造成，所以在精加工前要测量中径尺寸，并在刀具参数中进行补偿。
- 2) 螺距不正确。可能是螺纹参数不正确，要调整计算参数。
- 3) 牙型不正确。主要是由于车刀刃磨不正确，车刀装夹不正确或车刀磨损造成。

4) 表面粗糙度值大。主要原因为切削过程中产生积屑瘤、刀杆刚性不足、切削时产生振动、高速车螺纹时切削厚度太小或切屑向倾斜方向排出拉毛螺纹牙侧等。

5.4.2 单线梯形螺纹加工

1. 梯形螺纹车削程序设计

(1) 螺纹编程指令的选用

FANUC Series 0i Mate-TC 系统螺纹编程指令有 G32、G92、G76。其中 G32 是简单螺纹切削指令，编辑加工程序段多，一般不建议使用；G92 螺纹切削循环指令既可以采用直进式进刀方式也可通过增量值编程实现斜进式进刀方式；G76 螺纹切削复合循环指令采用斜进式进刀方式。由于切削方法的不同，则编程方法不同。在梯形螺纹加工时要避免三刃切削，在操作使用上要仔细分析，采用合适的编辑指令加工出精度较高的零件。本文则采用 G92 螺纹切削循环指令通过增量值编程实现斜进式进刀方式来加工梯形螺纹。

(2) 程序设计

1) 程序设计利用 G92 螺纹切削循环指令通过调用子程序，采用增量值编程加工梯形螺纹。在调用子程序加工梯形螺纹时要注意：螺纹刀起刀点 X 方向值必须大于或等于螺纹大径加 2 倍牙型高度值，且粗、精车刀的起刀点应取值合理，要随梯形螺纹加工坐标的变化而变化。

2) 梯形螺纹在加工过程中每刀的切削量不能过大，切削次数较多，并且为防止螺纹车刀三面切削，在加工过程中需分层切削，同时采用斜进法来进行车削，以减小切削力。在同一层上要车削几刀才能满足其宽度要求，通过对梯形螺纹刀刀头宽度的精准测量，再使用增量坐标编制子程序，在加工程序中多次调用子程序车削出梯形螺纹。以车梯形螺纹为例

(图 5-23)：对梯形螺纹进行分层车削，对参数背吃刀量赋初值，通过牙型结构的三角函数关系计算每层的纵向切削宽度及横向背吃刀量，车削时每层均采用斜进法分 10 刀切削，每次纵向切削循环增量等于该层计算值减刀宽除以 10，横向切削循环增量等于该层计算值减刀宽除以 10，切削后反向拓宽，纵向切削循环增量等于该层计算值减刀宽除以 10，因背吃刀量不大，切削力较小，一般不会产生振动和扎刀现象(图 5-24)。最后左右借刀光整加工(刀头宽 1.08mm)。

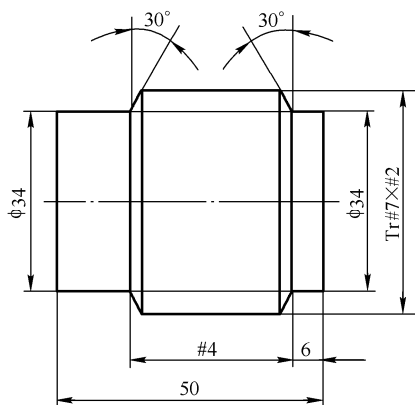


图 5-23 梯形螺纹示意图

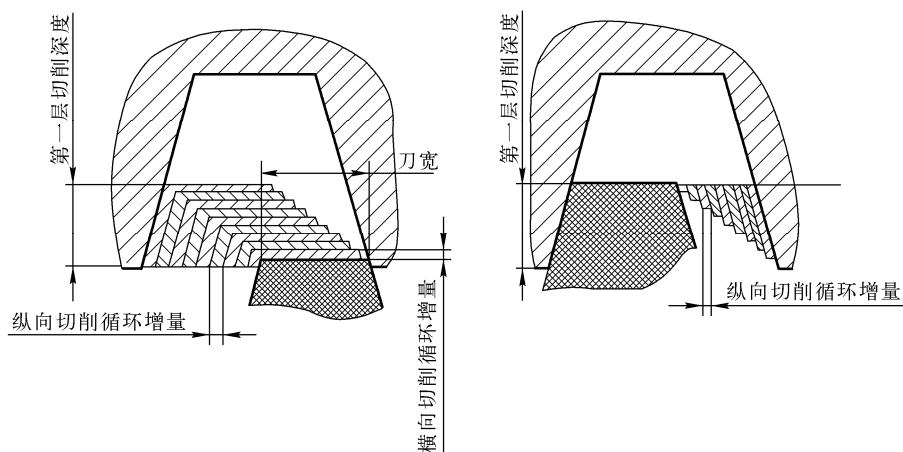


图 5-24 切削循环示意图

(3) 采用用户宏指令调用加工零件

1) 参数说明 (使用第Ⅱ类自变量赋值):

#1; 梯形螺纹刀宽

#2; 梯形螺纹螺距

#3; 梯形螺纹牙型高度

#4; 梯形螺纹长度

#5; 循环次数

#6; 梯形螺纹牙型半角

#7; 梯形螺纹公称直径

#10; X 向切削循环初始背吃刀量

#11; 累计背吃刀量

#12; 对应 X 向切削循环总背吃刀量在 Z 向变化量

#13; Z 向切削循环总量

#14; Z 向切削循环增量

#15; 梯形螺纹起点在工件坐标系的 Z 坐标

2) 主程序如下:

O0001;

N×× G98 M03 S400 T0101 F100; 零件加工工艺设计

N×× G00 X100 Z150; 刀具起刀点

N×× ...;

N×× G65 PO0002 A_B_I_J_K_I_I_I_K_; 调用梯形螺纹加工用户宏程序

N×× ...;

N×× G00 X100 Z150; 刀具返回起刀点

N×× M05; 主轴停止

N×× M30; 程序结束并返回程序开头

3) 用户宏程序如下:

O0002;

N×× IF [#2LT5] THEN[#3=#2/2+0.25]; 梯形螺纹牙型高度赋值

N×× IF [#2LT12] THEN[#3=#2/2+0.5]; 梯形螺纹牙型高度赋值

N×× IF [#2GT12] THEN[#3=#2/2+1]; 梯形螺纹牙型高度赋值

N×× IF [#2LE5] THEN #10=2; 横向切削循环初始背吃刀量 (直径值)

N×× IF [#2GT5] THEN #10=3; 横向切削循环初始背吃刀量 (直径值)

N×× G00 X[#7+2*#3] Z[#15+#2]; 螺纹起刀点定位

N×× #11=#10; 横向累计背吃刀量计算

N×× #12=#10/2*TAN[#6]; 对应横向切削循环背吃刀量在纵向变化量

N×× #13=0.634*#2-#12; 纵向切削循环量计算

N×× #13=#13-#12; 纵向切削循环总量

N×× #14=[#13-#1]/#5; 纵向切削循环增量

N×× M98 P100003; 调用程序加工梯形螺纹

N×× M98 P100004; 调用程序加工梯形螺纹

N5 #10=#10-0.5; 横向背吃刀量计算

N×× #11=#11+#10; 横向累计背吃刀量计算

N×× IF [#11GT2*#3] GOTO10; 横向背吃刀量判别

N×× #12=#10/2*TAN[#6]; 对应横向切削循环背吃刀量在纵向第 n 次变化量

N×× #13=#13-#12; 纵向第 n 次切削循环总量

N×× #14=[#13-#1]/#5; 纵向切削循环增量

N×× M98 P100003; 调用程序加工梯形螺纹

N×× M98 P100004; 调用程序加工梯形螺纹

N×× IF [#11LT2*#3] GOTO5; 横向背吃刀量判别

N10 #10=2*#3-#11+#10; 横向最后一次切削循环背吃刀量赋值

N×× #12=#10/2*TAN[#6]; 对应横向最后一次切削循环背吃刀量在纵向变化量

N×× #13=#13-#12; 纵向最后一次切削循环总量

N×× #14=[#13-#1]/#5; 纵向切削循环增量

N×× M98 P100003; 调用程序加工梯形螺纹

N×× M98 P100004; 调用程序加工梯形螺纹

N×× M98 P0005; 调用程序修梯形螺纹牙侧

```
N×× G00 X150 Z100; 刀具返回起刀点
N×× M99; 子程序结束
O0003;
N×× G01 U-[#10/#5] W-[#14-0.03] F80; 螺纹起刀点定位
N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削
N×× G00 U0; 取消固定循环
N×× M99; 子程序结束
O0004;
N×× G01 U0 W[[#13-#1]/#5-0.03] F80; 螺纹起刀点定位
N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削
N×× G00 U0; 取消固定循环
N×× M99; 子程序结束
O0005;
N×× G01 W0.02 F80; 精加工螺纹右侧起刀点定位
N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削
N×× G01 W-[0.366*#2-0.27-#1] F80; 精加工螺纹左侧起刀点定位
N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削
N×× G01 W-0.02 F80; 精加工螺纹左侧起刀点定位
N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削
N×× G00 U0; 取消固定循环
N×× M99; 子程序结束
```

2. 宏程序实例

编制一个车削梯形螺纹零件的程序，使用变量编制该零件加工的宏程序。

1) 程序设计利用 G92 螺纹切削循环指令通过调用子程序，采用增量值编程加工梯形螺纹。在调用子程序加工梯形螺纹时要注意：螺纹刀起刀点 X 方向值必须大于或等于螺纹大径加 2 倍牙型高度值，且粗、精车刀的起刀点应取值合理，要随梯形螺纹加工坐标的变化而变化。

2) 梯形螺纹在加工过程中每刀的切削量不能过大，切削次数较多，并且为防止螺纹车刀三面切削，在加工过程中需分层切削，同时采用斜进法来进行车削，以减小切削力，在同一层上要车削几刀才能满足其宽度要求，通过对梯形螺纹刀的刀头宽度的精准测量，再使用增量坐标编制子程序，在加工程序中多次调用子程序车削出梯形螺纹。以车 Tr42×12 梯形螺纹为例（图 5-25）：对梯形螺纹进行分层车削，对参数背吃刀量赋初值，通过牙型结构的三角函数关系计算每层的纵向切削宽度及横向背吃刀量，车削时每层均采用斜进法分 10 刀切削，每次纵向切

削循环增量等于该层计算值减刀宽除以 10, 横向切削循环增量等于该层计算值减刀宽除以 10, 切削后反向拓宽, 纵向切削循环增量等于该层计算值减刀宽除以 10, 因背吃刀量不大, 切削力较小, 一般不会产生振动和扎刀现象。最后左右借刀光整加工 (刀头宽 1.08mm)。

① 主程序如下:

O0001;

N×× G98 M03 S400 T0101

F100; 零件加工工艺设计

N×× G00 X100 Z150; 刀具起
刀点

N×× ...;

N×× G65 PO0002 A1.08 B6 I30 J10 K15 I42 I1 K-6; 调用梯形螺纹加工用户宏程序

N×× ...;

N×× G00 X100 Z150; 刀具返回起刀点

N×× M05; 主轴停止

N×× M30; 程序结束并返回程序开头

② 用户宏程序如下:

O0002;

N×× IF [#2LT5] THEN[#3=#2/2+0.25]; 梯形螺纹牙型高度赋值

N×× IF [#2LT12] THEN[#3=#2/2+0.5]; 梯形螺纹牙型高度赋值

N×× IF [#2GT12] THEN[#3=#2/2+1]; 梯形螺纹牙型高度赋值

N×× IF [#2LE5] THEN #10=2; 横向切削循环初始背吃刀量 (直径值)

N×× IF [#2GT5] THEN #10=3; 横向切削循环初始背吃刀量 (直径值)

N×× G00 X[#7+2*#3] Z[#15+#2]; 螺纹起刀点定位

N×× #11=#10; 横向累计背吃刀量计算

N×× #12=#10/2*TAN[#6]; 对应横向切削循环背吃刀量在纵向变化量

N×× #13=0.634*#2-#12; 纵向切削循环量计算

N×× #13=#13-#12; 纵向切削循环总量

N×× #14=[#13-#1]/#5; 纵向切削循环增量

N×× M98 P100003; 调用程序加工梯形螺纹

N×× M98 P100004; 调用程序加工梯形螺纹

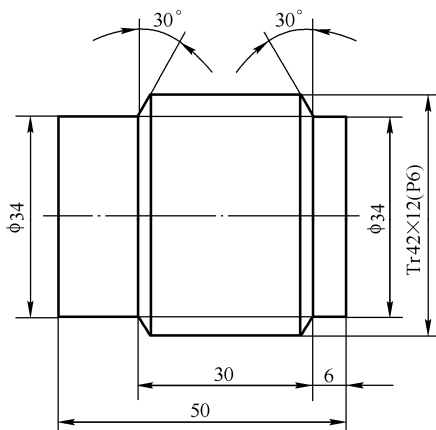


图 5-25 单线梯形螺纹零件实例

N5 #10=#10-0.5; 横向背吃刀量计算
N×× #11=#11+#10; 横向累计背吃刀量计算
N×× IF [#11GT2*#3] GOTO10; 横向背吃刀量判别
N×× #12=#10/2*TAN[#6]; 对应横向切削循环背吃刀量在纵向第 n 次变化量
N×× #13=#13-#12; 纵向第 n 次切削循环总量
N×× #14=[#13-#1]/#5; 纵向切削循环增量
N×× M98 P100003; 调用程序加工梯形螺纹
N×× M98 P100004; 调用程序加工梯形螺纹
N×× IF [#11LT2*#3] GOTO5; 横向背吃刀量判别
N10 #10=2*#3-#11+#10; 横向最后一次切削循环背吃刀量赋值
N×× #12=#10/2*TAN[#6]; 对应横向最后一次切削循环背吃刀量在纵向变化量
N×× #13=#13-#12; 纵向最后一次切削循环总量
N×× #14=[#13-#1]/#5; 纵向切削循环增量
N×× M98 P100003; 调用程序加工梯形螺纹
N×× M98 P100004; 调用程序加工梯形螺纹
N×× M98 P0005; 调用程序修梯形螺纹牙侧
N×× G00 X150 Z100; 刀具返回起刀点
N×× M99; 子程序结束
O0003;
N×× G01 U-[#10/#5] W-[#14-0.03] F80; 螺纹起刀点定位
N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削
N×× G00 U0; 取消固定循环
N×× M99; 子程序结束
O0004;
N×× G01 U0 W[[#13-#1]/#5-0.03] F80; 螺纹起刀点定位
N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削
N×× G00 U0; 取消固定循环
N×× M99; 子程序结束
O0005
N×× G01 W0.02 F80; 精加工螺纹右侧起刀点定位
N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削
N×× G01 W-[0.366*#2-0.27-#1] F80; 精加工螺纹左侧起刀点定位
N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削

N×× G01 W-0.02 F80; 精加工螺纹左侧起刀点定位

N×× G92 U-[2*#3] W-[#4+3*#2] F#2; 螺纹切削

N×× G00 U0; 取消固定循环

N×× M99; 子程序结束

5.4.3 多线梯形螺纹加工

1. 多线梯形螺纹车削程序设计

多线梯形螺纹车削程序设计以单线梯形螺纹车削程序为参考，同样由子程序及宏指令完成，所用的参数都可根据零件图样的要求进行赋值。程序的适应性、通用性强。

2. 程序设计流程

1) 以同一背吃刀量层上整个切削槽宽为转移条件来完成螺纹某一层加工。刀具的纵向移动通过不同的起刀点，采用相对坐标编程来完成同一背吃刀量层上整个槽宽的切削加工。程序在每层横向赋值时通过函数计算得到该层的槽宽，槽宽在纵向分十次进刀实现。

2) 以螺纹的牙型高度为转移条件来完成螺纹的分层加工。每一层的螺旋槽加工完后通过计算完成下一层的背吃刀量赋值，最终通过计算完成牙型高度赋值。

3) 以螺纹的线数为加工转移条件来完成多线螺纹的加工。程序中每加工完一条螺旋线后，刀具会沿轴向移动一个螺距再完成下一条螺旋线的加工。

3. 宏程序实例（图 5-26）

（1）参数说明（使用第Ⅱ类自变量赋值）

#1; 梯形螺纹刀宽

#2; 梯形螺纹螺距

#3; 梯形螺纹牙型高度

#4; 梯形螺纹长度

#5; 循环次数

#6; 梯形螺纹牙型半角

#7; 梯形螺纹公称直径

#8; 螺纹导程

#9; 螺纹线数

#10; 横向切削循环初始背吃刀量

#11; 横向累计背吃刀量

#12; 对应横向切削循环总背吃刀量在纵向变化量

#13; 纵向切削循环总量

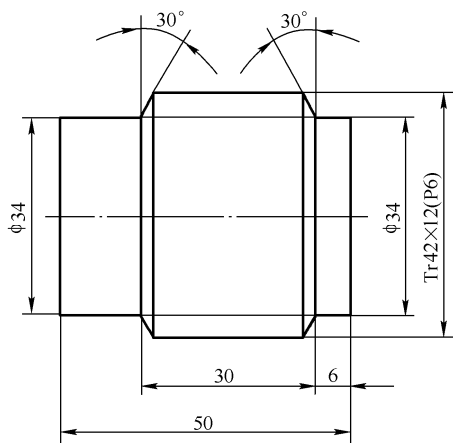


图 5-26 双线梯形螺纹零件实例

#14; 纵向切削循环增量

#15; 升速起刀段计算

#16; 已加工线数计算

(2) 主程序

O0001;

N×× G98 M03 S400 T0101 F100; 零件加工工艺设计

N×× G00 X100 Z150; 刀具起刀点

N×× ...

N×× G65 PO0002 A1.08 B6 I30 J10 K15 I42 J12 I I I0; 调用梯形螺纹加工
用户宏程序

N×× ...

N×× G00 X100 Z150; 刀具返回起刀点

N×× M05; 主轴停止

N×× M30; 程序结束并返回程序开头

(3) 用户宏程序

O0002;

N×× IF [#2LT5] THEN [#3=#2/2+0.25]; 梯形螺纹牙型高度赋值

N×× IF [#2LT12] THEN [#3=#2/2+0.5]; 梯形螺纹牙型高度赋值

N×× IF [#2GT12] THEN [#3=#2/2+1]; 梯形螺纹牙型高度赋值

N×× #9=#8/#2; 螺纹线数计算

N×× #15=2*#8; 升速起刀段计算

N01 G00 X[#7+2*#3] Z#15; 螺纹起刀点定位

N×× IF [#2LE5] THEN #10=2; 横向切削循环初始背吃刀量 (直径值)

N×× IF [#2GT5] THEN #10=3; 横向切削循环初始背吃刀量 (直径值)

N×× #11=#10; 横向累计背吃刀量计算

N×× #12=#10/2*TAN[#6]; 对应横向切削循环背吃刀量在纵向变化量

N×× #13=0.634*#2-#12; 纵向切削循环量计算

N×× #13=#13-#12; 纵向切削循环总量

N×× #14=[#13-#1]/#5; 纵向切削循环增量

N×× M98 P100003; 调用程序加工梯形螺纹

N×× M98 P100004; 调用程序加工梯形螺纹

N5 #11=#11+#10; 横向累计背吃刀量计算

N×× IF [#11GT2*#3] GOTO10; 横向背吃刀量判别

N×× #12=#10/2*TAN[#6]; 对应横向切削循环背吃刀量在纵向第 n 次变化量

N×× #13=#13-#12; 纵向第 n 次切削循环总量

```
N×× #14=[#13-#1]/#5; 纵向切削循环增量
N×× M98 P100003; 调用程序加工梯形螺纹
N×× M98 P100004; 调用程序加工梯形螺纹
N×× IF [#11LT2*#3] GOTO5; 横向背吃刀量判别
N10 #10=2*#3-#11+#10; 横向最后一次切削循环背吃刀量赋值
N×× #12=#10/2*TAN[#6]; 对应横向最后一次切削循环背吃刀量在纵向变
      化量
N×× #13=#13-#12; 纵向最后一次切削循环总量
N×× #14=[#13-#1]/#5; 纵向切削循环增量
N×× M98 P100003; 调用程序加工梯形螺纹
N×× M98 P100004; 调用程序加工梯形螺纹
N×× M98 P0005; 调用程序修梯形螺纹牙侧
N×× #15=#15+#2; 第 n 条螺旋线纵向起点
N×× #9=#9-1; 未完成线数计算
N×× #16=#16+1; 完成线数计算
N×× IF [#9GT0] GOTO 1; 线数条件转移
N×× G00 X150 Z100; 刀具返回起刀点
N×× M05; 主轴停止
N×× M99; 子程序结束
O0003;
N×× G01 U-[#10/#5] W-[#14-0.03] F80; 螺纹起刀点定位
N×× G92 U-[2*#3] W-[#4+[3+#16/2]*#8] F#8; 螺纹切削
N×× G00 U0; 取消固定循环
N×× M99; 子程序结束
O0004;
N×× G01 U0 W[[#13-#1]/#5-0.03] F80; 螺纹起刀点定位
N×× G92 U-[2*#3] W-[#4+[3+#16/2]*#8] F#8; 螺纹切削
N×× G00 U0; 取消固定循环
N×× M99; 子程序结束
O0005;
N×× G01 W0.02 F80; 精加工螺纹右侧起刀点定位
N×× G92 U-[2*#3] W-[#4+[3+#16/2]*#8] F#8; 螺纹切削
N×× G01 W-[0.366*#2-0.27-#1] F80; 精加工螺纹左侧起刀点定位
N×× G92 U-[2*#3] W-[#4+[3+#16/2]*#8] F#8; 螺纹切削
N×× G01 W-0.02 F80; 精加工螺纹左侧起刀点定位
```

N×× G92 U-[2*#3] W-[#4+[3+#16/2]*#8] F#8; 螺纹切削

N×× G00 U0; 取消固定循环

N×× M99; 子程序结束

5.4.4 变螺距螺纹加工

1. 变螺距螺纹概述

变螺距螺纹指螺纹的螺距（导程）不是定值，而是沿着螺旋线方向逐渐变大或减小，有等槽宽变螺距螺纹和等牙宽变螺距螺纹两种形式，如图 5-27、图 5-28 所示。

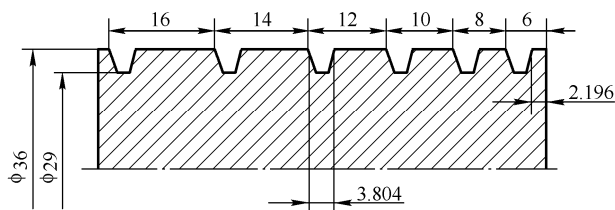


图 5-27 等槽宽变螺距螺纹

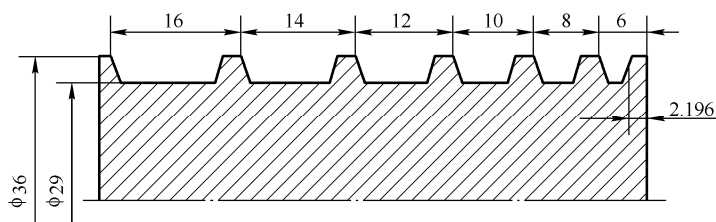


图 5-28 等牙宽变螺距螺纹

2. 变螺距螺纹车削程序设计

变螺距螺纹车削程序设计以单线梯形螺纹车削程序为参考，同样由子程序及宏指令完成，所用的参数都可根据零件图样的要求进行赋值。程序的适应性、通用性强。

3. 程序设计流程

1) 程序设计利用 G32 螺纹切削指令通过调用子程序，采用增量值编程加工变螺距螺纹。在调用子程序加工变螺距螺纹时同样要注意：螺纹刀起刀点 X 方向值必须大于或等于螺纹大径加 2 倍牙型高度值，且粗、精车刀的起刀点取值应合理，要随变螺距螺纹加工坐标的变化而变化。

2) 变螺距螺纹在加工过程中每刀的切削量也不能过大，切削次数较多，并且为防止螺纹车刀三面切削，在加工过程中需分层切削，同时采用斜进法来进行车削，以减小切削力，在同一层上要车削几刀才能满足其宽度要求，通过对梯形螺

纹刀的刀头宽度的精准测量，再使用增量坐标编制子程序，在加工程序中多次调用子程序车削出梯形螺纹。以等槽宽变螺距螺纹为例（图 5-27），对梯形螺纹进行分层车削，对参数背吃刀量赋初值，通过牙型结构的三角函数关系计算每层的纵向切削宽度及横向背吃刀量，车削时每层均采用斜进法分 10 刀切削，每次纵向切削循环增量等于该层计算值减刀宽除以 10，横向切削循环增量等于该层计算值减刀宽除以 10，切削后反向拓宽，纵向切削循环增量等于该层计算值减刀宽除以 10，因背吃刀量不大，切削力较小，一般不会产生振动和扎刀现象。最后左右借刀光整加工（刀头宽 1.08mm）。

4. 宏程序实例

在数控车床上加工如图 5-27 所示的等槽宽变螺距螺纹。

分析：从图 5-27 中可以看出该丝杠的牙型为梯形，大径为 36mm，小径为 29mm，第一个导程为 6mm，每转递增 2mm，为等槽宽变螺距螺纹。加工该螺纹的方法与加工梯形螺纹相似，刀具的选用与加工梯形螺纹相同。

采用用户宏指令调用加工零件，具体加工参考程序如下。

（1）参数说明（使用第 II 类自变量赋值）

#1：梯形螺纹刀宽

#2：变导程螺纹初始螺距

#3：梯形螺纹牙型高度

#4：每转螺距递增量

#5：循环次数

#6：梯形螺纹牙型半角

#7：变导程螺纹直径

#8：变导程螺纹终止螺距

#10：X 向切削循环初始背吃刀量

#11：累计背吃刀量

#12：对应 X 向切削循环总背吃刀量在 Z 向变化量

#13：Z 向切削循环总量

#14：Z 向切削循环增量

#15：变导程螺纹加工螺距变量

#17：变导程螺纹切削循环次数计算

（2）主程序

O0001；

N×× G98 M03 S400 T0101 F100；零件加工工艺设计

N×× G00 X100 Z150；刀具起刀点

N×× ...

N×× G65 PO0002 A1.08 B6 I2 J10 K15 D36 E16 Q0; 调用梯形螺纹加工用户宏程序

N×× ...

N×× G00 X100 Z150; 刀具返回起刀点

N×× M05; 主轴停止

N×× M30; 程序结束并返回程序开头

(3) 用户宏程序

O0002;

N×× IF [#2LT5] THEN[#3=#2/2+0.25]; 梯形螺纹牙型高度赋值

N×× IF [#2LT12] THEN[#3=#2/2+0.5]; 梯形螺纹牙型高度赋值

N×× IF [#2GT12] THEN[#3=#2/2+1]; 梯形螺纹牙型高度赋值

N×× IF [#2LE5] THEN #10=2; 横向切削循环初始背吃刀量 (直径值)

N×× IF [#2GT5] THEN #10=3; 横向切削循环初始背吃刀量 (直径值)

N×× G00 X[#7+2*#3] Z[#2-#4]; 螺纹起刀点定位

N×× #11=#10; 横向累计背吃刀量计算

N×× #12=#10/2*TAN[#6]; 对应横向切削循环背吃刀量在纵向变化量

N×× #13=0.634*#2-#12; 纵向切削循环量计算

N×× #13=#13-#12; 纵向切削循环总量

N×× #14=[#13-#1]/#5; 纵向切削循环增量

N×× M98 P100003; 调用程序加工梯形螺纹

N×× M98 P100004; 调用程序加工梯形螺纹

N5 #10=#10-0.5; 横向背吃刀量计算

N×× #11=#11+#10; 横向累计背吃刀量计算

N×× IF [#11GT2*#3] GOTO10; 横向背吃刀量判别

N×× #12=#10/2*TAN[#6]; 对应横向切削循环背吃刀量在纵向第 n 次变化量

N×× #13=#13-#12; 纵向第 n 次切削循环总量

N×× #14=[#13-#1]/#5; 纵向切削循环增量

N×× M98 P100003; 调用程序加工梯形螺纹

N×× M98 P100004; 调用程序加工梯形螺纹

N×× IF [#11LT2*#3] GOTO5; 横向背吃刀量判别

N10 #10=2*#3-#11+#10; 横向最后一次切削循环背吃刀量赋值

N×× #12=#10/2*TAN[#6]; 对应横向最后一次切削循环背吃刀量在纵向变化量

N×× #13=#13-#12; 纵向最后一次切削循环总量

N×× #14=[#13-#1]/#5; 纵向切削循环增量

```
N×× M98 P100003; 调用程序加工梯形螺纹
N×× M98 P100004; 调用程序加工梯形螺纹
N×× M98 P0005; 调用程序修梯形螺纹牙侧
N×× G00 X150 Z100; 刀具返回起刀点
N×× M05; 主轴停止
N×× M99; 子程序结束
O0003;
N×× G01 U-[#10/#5] W-[#14-0.03] F80; 螺纹起刀点定位
N×× U-[2*#3]; 螺纹起刀点定位
N×× G65 P0006 B6 I2 E16; 调用变导程螺纹切削
N×× G00 U[2*#3]; 螺纹退刀点定位
N×× Z[#2-#4]; 螺纹退刀点定位
N×× #17=#17+1; 变导程螺纹切削循环次数计算
N×× W-[[#14-0.03]*#17]; 螺纹起刀点定位
N×× M99; 子程序结束
O0004;
N×× G01 U0 W[[#13-#1]/#5-0.03] F80; 螺纹起刀点定位
N×× U-[2*#3]; 螺纹起刀点定位
N×× G65 P0006 B6 I2 E16; 调用变导程螺纹切削
N×× G00 U[2*#3]; 螺纹退刀点定位
N×× Z[#2-#4]; 螺纹退刀点定位
N×× #17=#17+1; 变导程螺纹切削循环次数计算
N×× W[[#13-#1]/#5-0.03]*#17]; 螺纹起刀点定位
N×× M99; 子程序结束
O0005;
N×× G01 W0.02 F80; 精加工螺纹右侧起刀点定位
N×× U-[2*#3]; 螺纹起刀点定位
N×× G65 P0006 B6 I2 E16; 调用变导程螺纹切削
N×× G00 U[2*#3]; 螺纹退刀点定位
N×× Z[#2-#4]; 螺纹起刀点定位
N×× G01 W-[0.366*#2-0.27-#1] F80; 精加工螺纹左侧起刀点定位
N×× U-[2*#3]; 螺纹起刀点定位
N×× G65 P0006 B6 I2 E16; 调用变导程螺纹切削
N×× G00 U[2*#3]; 螺纹退刀点定位
N×× Z[#2-#4]; 螺纹起刀点定位
```

```
N×× G01 W-[0.366*#2-0.29-#1] F80; 精加工螺纹左侧起刀点定位
N×× U-[2*#3]; 螺纹起刀点定位
N×× G65 P0006 B6 I2 E16; 调用变导程螺纹切削
N×× G00 U[2*#3]; 螺纹退刀点定位
N×× Z[#2-#4]; 螺纹起刀点定位
N×× M99; 子程序结束
O0006;
N×× #15=#2-#4; 变导程螺纹螺距赋初值
N15 G32 W[-#15] F#15; 变导程螺纹加工
N×× #15=#15+#4; 变导程螺纹螺距运算
N×× IF [#15LE#8] GOTO15; 循环体判断
N×× M99; 子程序结束
```

用该程序加工出来的为等槽宽变螺距螺纹，如果要加工等牙宽变螺距螺纹，可以先车出一条等槽宽的变螺距螺纹，然后通过改变螺纹切削起点和螺距的方法来实现赶刀，从而改变槽宽，即可以得到等牙宽变螺距螺纹。

第 6 章 数控铣床宏程序应用实例

6.1 规则形状加工

6.1.1 实例一圆柱体加工

1. 外圆柱体加工

(1) 粗加工 (立铣刀、由上而下)

1) 分析: 如图 6-1 所示, 外圆柱体直径 $\#1$, 高度 $\#3$, 立铣刀直径 $\#2$, 每次背吃刀量 $\#6$, 背吃刀量取值时保证 $\#3/\#6$ 是整数值, 即切削次数是正整数; 采用螺旋切削, 刀具顺时针螺旋切削一整圈深度下降一个 $\#6$ 值, 直至总深度 $\#3$, 再水平圆弧整圆切削一次保证底部平整。

2) 变量及说明:

$\#1=[A]$; 外圆柱体直径

$\#2=[B]$; 刀具直径

$\#3=[C]$; 外圆柱体高度

$\#6=[K]$; 每次背吃刀量递增量

$\#7=[D]$; 进给量

3) 宏程序如下:

① 无半径补偿方式编程。

%

O6111; 子程序号

G00 X[#1/2+#2] Y0; 快速定位到 X=圆柱体半径+刀具直径, Y0 处
Z5; 快速下降到 Z=5

G01 Z0 F[0.3*#7]; 直线切削到工件上平面

X[#1+#2]/2 F#7; 水平直线切削至螺旋切削起始点

$\#10=\#6$; 给每次背吃刀量赋初始值 $\#6$

WHILE[#10LE#3] DO1; 循环 1 设定条件: 每次背吃刀量不大于总深度

G02 I[-#1+#2]/2 Z-#10; 螺旋切削

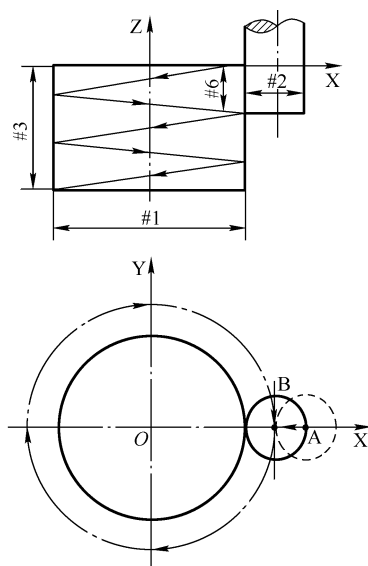


图 6-1 外圆柱体粗加工

#10=#10+#6; 每次背吃刀量递增一个#6 值

END1; 循环 1 结束

G02 I[-#1+#2]/2; 最后水平圆弧整圆切削一次

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

② 利用半径补偿指令编程。(#18: 圆弧切入切出半径)

%

O6111; 子程序号

G00 X[#1/2+#18] Y0; 快速定位到加工起始位置

Z0; 快速下降到 Z0

G41 D1 Y#18; 建立半径补偿

G03 X[#1/2] Y0 R#18 F#7; 圆弧切入

#10=#6; 给背吃刀量变量#10 赋初始值#6

WHILE[#10LE#3] DO1; 循环 1 设定条件: 背吃刀量变量#10 不大于#3

G02 I[-#1/2] Z-#10; 顺时针整圆螺旋铣削

#10=#10+#6; 背吃刀量变量累加一个深度增量#6

END1; 循环 1 结束

G02 I[-#1/2]; 顺时针整圆水平铣削, 铣平底面

G03 X[#1/2+#18] Y-#18 Z[2-#3] R#18; 螺旋切出

G00 Z50; 快速上升至 Z=50

G40 X0 Y0; 取消半径补偿

M99; 子程序结束

%

4) 主程序如下:

%

O111;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S400;

G65 P6111 A70 B16 C10 K2 D300 R10; 调用宏程序 O6111, 给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 精加工

1) 分析: 如图 6-2 所示, 外圆柱体直径 #1, 高度 #3, 立铣刀直径 #2; 圆弧切入切出保证表面粗糙度, 切入切出圆弧半径为 #18, 刀具先快速定位至 $X=\text{外圆柱体半径}+\text{切入切出圆弧半径}$, $Y0$ 处, 然后快速下降至总深度 #3, 再顺时针圆弧整圆切削, 最后圆弧切出。

2) 变量及说明:

#1=[A]; 外圆柱体直径

#2=[B]; 刀具直径

#3=[C]; 外圆柱体高度

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6112; 子程序号

G00 X[#1/2+#18] Y0; 快速定位至加工准备位置

Z-#3; 快速下降至总深度

G41 D1 Y#18; 建立半径补偿

G03 X[#1/2] Y0 R#18 F#7; 1/4 圆弧切入

G02 I[-#1/2]; 整圆顺时针圆弧切削

G03 X[#1/2+#18] Y-#18 R#18; 1/4 圆弧切出

G00 Z50; 快速上升至 $Z=50$

G40 X0 Y0; 取消半径补偿

M99; 子程序结束

%

4) 主程序如下:

%

O112;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6112 A70 C10 D1000 R10; 调用宏程序 O6112, 给相关变量赋

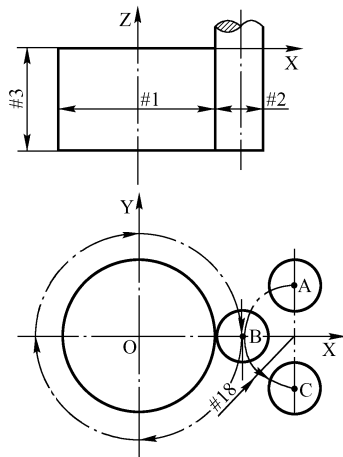


图 6-2 精加工

初始值

```
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

2. 内圆柱体加工

(1) 内圆柱体粗加工（立铣刀、由上而下）

1) 分析：如图 6-3 所示，内圆柱体直径#1，高度#3，孔壁精加工余量#4，孔底精加工余量#5。立铣刀直径#2，每次背吃刀量#6，背吃刀量取值时保证#3/#6 是整数值，即切削次数是正整数。

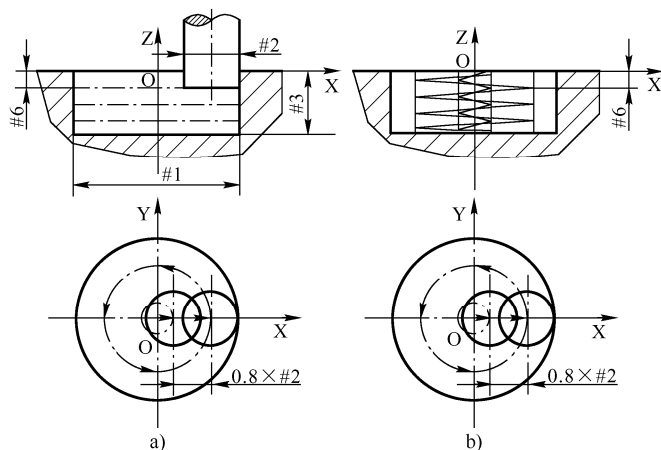


图 6-3 内圆柱体加工

a) 逐层铣削 b) 由内而外，逐个圆孔加工

有如下两种走刀路线：

① 如图 6-3a 所示，逐层铣削由内而外水平整圆圆弧逆时针切削，行间距 $\#10=0.8 \times \#2$ ，每次深度为#6，直至内圆柱直径-孔壁余量；然后抬刀 2mm，返回切削水平起始位置，再下深#6，继续进行由内而外水平整圆圆弧逆时针切削，一直到背吃刀量至总深度-孔底余量。

② 如图 6-3b 所示，由内而外采用螺旋切削，刀具先逆时针整圆螺旋切削一深度#6 值，每次深度为#6，直至背吃刀量至总深度-孔底余量，再水平圆弧整圆切削一次保证底部平整；返回 X0 Y0，抬刀到刚才切削起始高度，行间距 $\#10=0.8 \times \#2$ ，快速定位到下一轮螺旋切削水平起始位置，继续进行由内而外螺旋切削，一直到内圆柱直径+孔壁余量。

2) 变量及说明:

#1=[A]; 内圆柱体直径

#2=[B]; 刀具直径

#3=[C]; 内圆柱体高度

#4=[I]; 孔壁精加工余量

#5=[J]; 孔底精加工余量

#6=[K]; 每次背吃刀量递增量

#7=[D]; 进给量

3) 宏程序如下:

① 逐层铣削。

%

O6113; 子程序号

#10=0.8*#2; 计算行间距#10=0.8×刀具直径

#20=#1/2-#4-#2/2; 计算半径方向刀具中心最大极限尺寸

#21=#3-#5; 计算深度最大极限尺寸

#11=FIX[#20/#10]; 计算半径方向切削次数

#12=FIX[#21/#6]; 计算深度方向切削次数

G00 X0 Y0; 快速定位到 X0 Y0

Z2; 快速定位至 Z=2

#14=0; 给深度切削次数计数器赋初始值为 0

WHILE[#14LE#12] DO2; 循环 2 设定条件: 计数器#14 不大于计算值#12

#26=#6*[#14+1]; 背吃刀量#26 等于#6×[计数器值+1]

IF[#26GT#21] THEN#26=#21; 如果背吃刀量#26 大于计算值#21, 则#26=#21,
即背吃刀量#26 最大值只能是计算值#21

G01 Z-#26 F[0.3*#7]; 直线切削至深度#26

#13=0; 给水平切削次数计数器赋初始值为 0

WHILE[#13LE#11] DO1; 循环 1 设定条件: 水平切削计数器#13 不大于计
算值#11

#24=0.3*#10+#10*#13; 计算刀具中心水平 X 坐标值

IF[#24GT#20] THEN#24=#20; 如果刀具中心 X 坐标值#24 大于计算值#20,
则#24=#20, 即背吃刀量#26 最大值只能是#20

G01 X#24 F#7; 直线切削至 X=#24

G03 I-#24; 逆时针整圆圆弧切削

#13=#13+1; 水平切削计数器#13 累加 1 计数

END1; 循环 1 结束

G01 Z[2-#26]; 刀具抬高 2mm
G00 X0 Y0; 快速返回至 X0 Y0
#14=#14+1; 深度计数器#14 累加 1 计数
END2; 循环 2 结束
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
%
② 由内而外, 逐个圆孔加工。
%
O6114; 子程序号
#10=0.8*#2; 计算行间距#10=0.8×刀具直径
#20=#1/2-#4-#2/2; 计算半径方向刀具中心最大极限尺寸
#21=#3-#5; 计算深度最大极限尺寸
#11=FIX[#20/#10]; 计算半径方向切削次数
#12=FIX[#21/#6]; 计算深度方向切削次数
#13=0; 给水平切削次数计数器赋初始值为 0
WHILE[#13LE#11] DO1; 循环 1 设定条件: 水平切削计数器#13 不大于计算
值#11
#24=0.3*#10+#10*#13; 计算刀具中心水平 X 坐标值
IF[#24GT#20] THEN#24=#20; 如果刀具中心 X 坐标值#24 大于计算值#20,
则#24=#20, 即背吃刀量#26 最大值只能是#20
G00 X#24 Y0; 快速定位到水平 X 坐标等于#24, Y0 处
Z2; 快速下降至 Z=2
#14=0; 给深度切削次数计数器赋初始值为 0
WHILE[#14LE#12] DO2; 循环 2 设定条件: 计数器#14 不大于计算值#12
#26=#6*[#14+1]; 背吃刀量#26 等于#6×[计数器值+1]
IF[#26GT#21] THEN#26=#21; 如果背吃刀量#26 大于计算值#21, 则#26=#21,
即背吃刀量#26 最大值只能是计算值#21
G03 I-#24 Z-#26 F#7; 整圆螺旋切削, 深度至-#26
#14=#14+1; 深度计数器#14 累加 1 计数
END1; 循环 1 结束
G03 I-#24; 最后水平整圆圆弧切削一次
G00 X0 Y0; 快速定位至 X0 Y0
Z5; 上升到 Z=5
#13=#13+1; 水平切削计数器#13 累加 1 计数

END2; 循环 2 结束

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O113;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S400;

G65 P6113 A70 B16 C10 I0.4 J0.3 K2 D300 R10; 调用宏程序 O6113,
给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 内圆柱体孔底精加工

1) 分析: 如图 6-4 所示, 内圆柱体直径#1, 高度#3, 孔壁精加工余量#4, 立铣刀直径#2; 刀具先快速定位至 (0, 0) 处, 然后快速下降至 Z=2-#3 离底面 2mm 处, 直线切削到总深度, 水平由内而外逐圈逆时针圆弧整圆切削, 行间距 #10=0.8*#2, 直至内孔直径-孔壁余量。

2) 变量及说明:

#1=[A]; 内圆柱体直径

#2=[B]; 刀具直径

#3=[C]; 内圆柱体高度

#4=[I]; 孔壁精加工余量

#7=[D]; 进给量

3) 宏程序如下:

%

O6115; 子程序号

#10=0.8*#2; 计算行间距#10=0.8×刀具直径

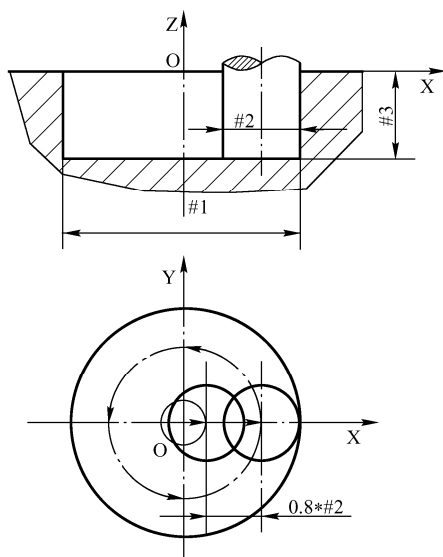


图 6-4 孔底精加工

#20=#1/2-#4-#2/2; 计算半径方向刀具中心最大极限尺寸

#11=FIX[#20/#10]; 计算半径方向切削次数

G00 X0 Y0; 快速定位到 X0 Y0

Z[2-#3]; 快速下降至离底面 2mm 处

G01 Z-#3 F[0.3*#7]; 直线切削到底面

#13=0; 给水平切削次数计数器赋初始值为 0

WHILE[#13LE#11] DO1; 循环 1 设定条件: 水平切削计数器#13 不大于计算值#11

#24=0.3*#10+#10*#13; 计算刀具中心水平 X 坐标值

IF[#24GT#20] THEN#24=#20; 如果刀具中心 X 坐标值#24 大于计算值#20, 则#24=#20, 即背吃刀量#26 最大值只能是#20

G01 X#24 F#7; 直线切削到水平 X 坐标等于#24, Y0 处

G03 I-#24; 逆时针整圆圆弧切削

#13=#13+1; 计数器累加 1

END1; 循环 1 结束

G00 Z50; 快速上升到 Z=50

G00 X0 Y0; 快速返回 X0 Y0

M99; 子程序结束

%

4) 主程序如下:

%

O115;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6115 A70 B16 C25 I0.3 D1000; 调用宏程序 O6115, 给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(3) 内圆柱体孔壁精加工

1) 分析: 如图 6-5 所示, 内圆柱体直径#1, 高度#3, 立铣刀直径#2; 圆弧切入切出保证表面粗糙度, 切入切出圆弧半径为#18, 刀具先快速定位至 X0 Y0 Z0

处, 建立半径补偿到 1/4 圆弧切入起始点, 圆弧切入半径为#18, 然后逆时针整圆螺旋切削, 每次深度为#6, 直至总深度#3, 再逆时针水平整圆圆弧切削, 保证孔底平整, 最后圆弧切出, 取消半径补偿, 抬刀。

2) 变量及说明:

#1=[A]; 内圆柱体直径

#2=[B]; 刀具直径

#3=[C]; 内圆柱体高度

#6=[K]; 每次背吃刀量递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6116; 子程序号

#12=FIX[#3/#6]; 计算深度切削次数

G00 X0 Y0; 快速定位到 X0 Y0

Z0; 快速下降至 Z0

G41 D1 X[#1/2-#18] Y-#18; 半径补偿至 1/4 圆弧切入起始点

G03 X[#1/2]Y0 R#18 Z-2 F#7; 圆弧切入

#14=0; 深度次数计数器#14 赋初始值为 0

WHILE[#14LE#12] DO2; 循环 2 设定条件: 计数器#14 不大于#12 值

#26=#6*[#14+1]; 任意次背吃刀量#26 计算

IF[#26GE#3] THEN#26=#3; 如果背吃刀量#26 值大于总深度, 则背吃刀量最大只能是总深度

G03 I[-#1/2] Z-#26; 螺旋切削至深度-#26

#14=#14+1; 计数器累加 1

END2; 循环 2 结束

G03 I[-#1/2]; 水平整圆圆弧切削, 保证底部平整

X[#1/2-#18] Y#18 Z[2-#26] R#18; 螺旋切出

G00 Z50; 快速抬刀到 Z=50

G40 X0 Y0; 取消半径补偿并快速回到 X0 Y0

M99; 子程序结束

%

4) 主程序如下:

%

O116;

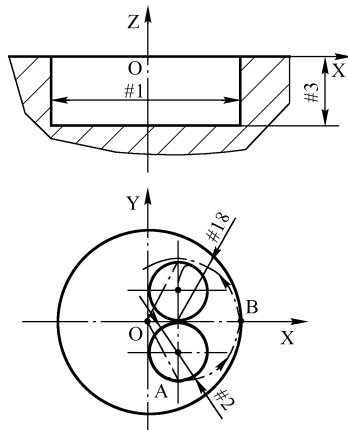


图 6-5 孔壁精加工

```

G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6116 A70 B16 C25 K6 D1000 R10; 调用宏程序 O6116, 给
                                         相关变量赋初始值

G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.1.2 实例二内外锥体

1. 外圆锥体粗加工

(1) 立铣刀(由上而下)

1) 分析: 如图 6-6 所示, 刀具每次的加工路线: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow B \rightarrow A$ 。

先计算出外圆锥小端半径, 刀具快速定位到外圆锥体右端中间距离 2mm 处, 下降至 $Z=5$, 计算每次背吃刀量, 相应圆柱半径, 然后下降至该次背吃刀量, 直线切削至相应圆柱半径+刀具半径处, 顺时针整圆圆弧切削, 水平返回刚才切削起始位置; 再下降一背吃刀量#6, 接着进行刚才的整圆等相关切削及退刀, 直至完成整个高度的切削。

2) 参数及说明:

#1=[A]; 锥体大端直径

#2=[B]; 刀具直径

#3=[C]; 锥体高度

#4=[I]; 锥度 1 : #4

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

3) 宏程序如下:

%

O6121; 子程序号

#5=[#1-#3/#4]/2; 计算小端半径

G00 X[(#1+#2)/2+2] Y0; 快速定位到刀尖与大端外轮廓 2mm 处

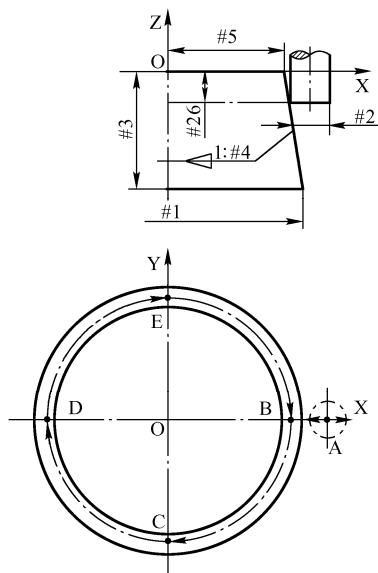


图 6-6 外圆锥体粗加工 (1)

Z5; 快速下降到 Z=5

#26=#6; 给深度赋初始值一个背吃刀量

WHILE[#26LE#3] DO1; 循环 1 设定条件: 背吃刀量#26 不大于总深度

#24=#5+#2/2+#26/#4/2; 计算相应水平 X 坐标值

Z=#26; 快速下降到背吃刀量#26

G01 X#24 F#7; 直线切削到该深度对应的 X 坐标值#24

G02 I=#24; 顺时针整圆圆弧切削

G00 Z[2-#26]; 抬刀 2mm

X[[#1+#2]/2+2] Y0; 返回刚才起始点水平位置

#26=#26+#6; 计算下一轮背吃刀量

END1; 循环 1 结束

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O121;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6121 A70 B16 C15 I5 D300 K1.5; 调用宏程序 O6121, 相应
变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 圆鼻刀

1) 分析: 如图 6-7a 所示, 刀具每次的加工路线: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow B \rightarrow A$ 。

因为锥度 = (圆锥大端直径 - 小端直径) / 圆锥高度, 即 $1 : \#4 = (D - d) / H$, 而
 $\text{TAN}[\#12] = H / [(D - d) / 2] = \#4 / 2$, 由此得 $\#12 = \text{ATAN}[\#4] / [2]$ 。

如图 6-7b 所示, 在 $\triangle ABC$ 中, $\angle ACB = \#12$, $AC = \#20$ 。

$AB = AC * \sin \angle ACB = \#20 * \sin[\#12]$, $BC = AC * \cos \angle ACB = \#20 * \cos[\#12]$ 。

P 点坐标与 A 点坐标的关系:

$DP = \text{刀具半径} - AC = \#2 / 2 - \#20$

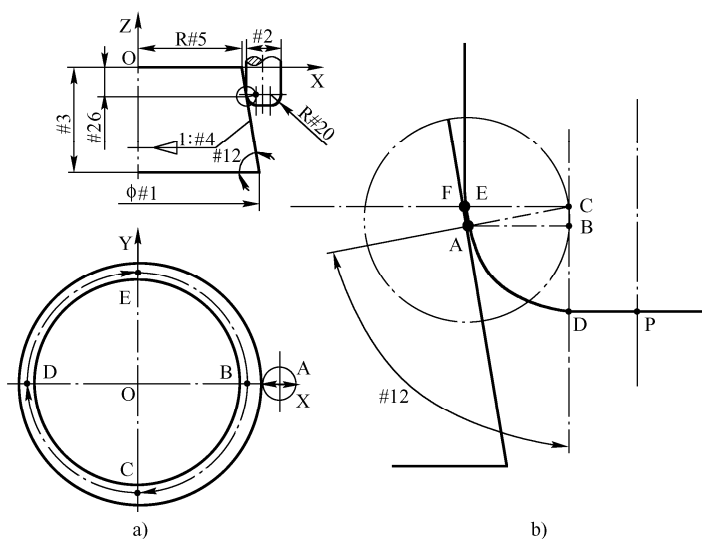


图 6-7 外圆锥体粗加工 (2)

a) 整体示意图 b) 局部放大图

$$X_P = X_A + AB + DP = X_A + \#2/2 - \#20 * [1 - \sin[\#12]]$$

$$BD = CD - BC = \#20 - \#20 * \cos[\#12]$$

$$Z_P = Z_A + BD = Z_A + \#20 * [1 - \cos[\#12]]$$

将上两 P 点 XY 坐标的计算公式, 运用到 O6121 中 A 点坐标计算中, 即可进行圆鼻刀的外圆锥体的加工。

2) 参数及说明:

#1=[A]; 锥体大端直径

#2=[B]; 刀具直径

#3=[C]; 锥体高度

#4=[I]; 锥度 1 : #4

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

#20=[T]; 刀尖角半径

3) 宏程序如下:

%

O6122; 子程序号

#12=ATAN[#4]/[2]; 计算大端锥角

#11=#20*[SIN[#12]-1]; 计算 EF 长度

#13=#20*[COS[#12]-1]; 计算 BD 长度

#5=[#1-#3/#4]/2; 计算圆锥小端半径

G00 X[[#1+#2]/2+2] Y0; 快速定位到刀尖与大端外轮廓 2mm 处

Z5; 快速下降到 Z=5

#26=#6; 给深度赋初始值一个背吃刀量

WHILE[#26LE#3] DO1; 循环 1 设定条件: 背吃刀量#26 不大于总深度

#24=#5+#2/2+#26/#4/2-#11; 计算相应水平 X 坐标值

Z[#13-#26]; 快速下降到背吃刀量#13-#26

G01 X#24 F#7; 直线切削到该深度对应的 X 坐标值#24

G02 I-#24; 顺时针整圆圆弧切削

G00 Z[2-#26]; 抬刀 2mm

X[[#1+#2]/2+2] Y0; 返回刚才起始点水平位置

#26=#26+#6; 计算下一轮背吃刀量

END1; 循环 1 结束

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O122;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6122 A70 B16 C15 I5

D300 K1.5 T3; 调用宏程序 O6122, 相

应变变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 外圆锥体精加工

(1) 立铣刀 (由下而上)

1) 分析: 如图 6-8 所示, 根据前面实例分析方法, 现在是精加工, 所以刀具首先快速定位到外圆锥体下方大锥体, 由

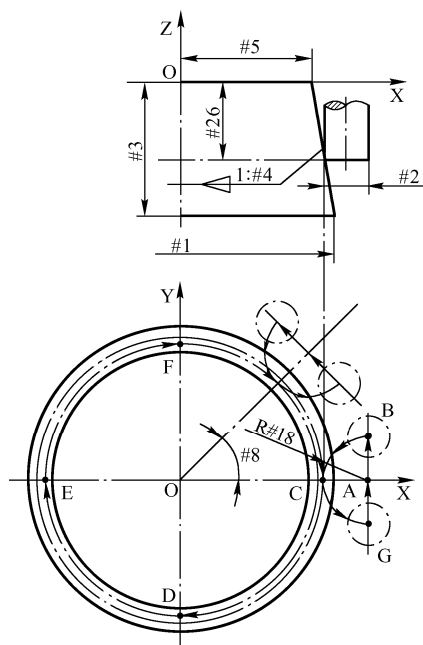


图 6-8 外圆锥体精加工 (1)

下而上逐层铣削。

采用螺旋切入，半径补偿，圆弧切出，加工路线轨迹： $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow C \rightarrow G \rightarrow A$ 。

2) 参数及说明：

#1=[A]；锥体大端直径

#2=[B]；刀具直径

#3=[C]；锥体高度

#4=[I]；锥度 1：#4

#6=[K]；背吃刀量增量

#7=[D]；进给量

#17=[Q]；角度递增量

#18=[R]；1/4 圆弧切入切出半径

3) 宏程序如下：

%

O6123；子程序号

#5=[#1-#3/#4]/2；计算小端半径

G00 X[#1/2+#2] Y0；快速定位到刀尖与大端外轮廓 2mm 处

Z-#3；快速下降到总深度

#26=#3；将总深度作为初始值赋给背吃刀量#26

#8=0；水平旋转角度赋初始值为 0 度

WHILE[#26GE0] DO1；循环 1 设定条件：背吃刀量#26 不小于 0

G68 X0 Y0 R#8；坐标系绕 X0Y0 旋转#8 角度

#24=#5+#26/#4/2；计算相应水平 X 坐标值

#14=#24+#18；计算 1/4 圆弧切入起始点 X 坐标

G00 G41 D1 X#14 Y#18；半径补偿定位到 1/4 圆弧切入起始点

G03 X#24 Y0 Z-#26 R#18 F#7；螺旋切入

G02 I-#24；顺时针整圆圆弧切削

G03 X#14 Y-#18 R#18；1/4 圆弧切出

G00 G40 X[#1/2+#2] Y0；取消半径补偿

#26=#26-#6；计算下一轮背吃刀量

#8=#8+#17；水平旋转角度累加一个角度递增量#17

G69；坐标系旋转取消

END1；循环 1 结束

G00 Z50；快速上升至 Z=50

M99；子程序结束

%

4) 主程序如下:

%

O123;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6123 A70 B16 C15 I5 D1000 K0.5 R10 Q10; 调用宏程序 O6123,
相应变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 圆鼻刀

1) 分析: 如图 6-9 所示, 这里刀具与上例不同, 用的是圆鼻刀, 刀具参考点仍然是刀具轴线顶点, 只是坐标计算有所区别, 其他方法相同。加工路线轨迹: A→B→C→D→E→F→C→G→A。

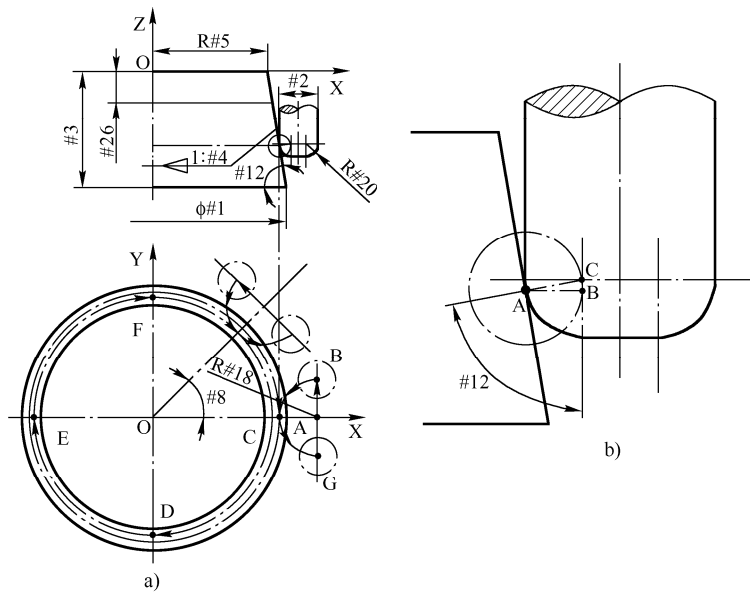


图 6-9 外圆锥体精加工 (2)

a) 整体示意图 b) 局部放大图

2) 参数及说明:

#1=[A]; 锥体大端直径

#2=[B]; 刀具直径

#3=[C]; 锥体高度

#4=[I]; 锥度 1 : #4

#6=[K]; 背吃刀量递增量

#7=[D]; 进给量

#17=[Q]; 角度递增量

#18=[R]; 1/4 圆弧切入切出半径

#20=[T]; 刀尖角半径

3) 宏程序如下:

%

O6124; 子程序号

#12=ATAN[#4]/[2]; 计算圆锥大端锥角

#11=#20*[SIN[#12]-1]; 计算刀具中心点与切削点 X 轴坐标值差值

#13=#20*[COS[#12]-1]; 计算刀具中心点与切削点 Z 轴坐标值差值

#5=[#1-#3/#4]/2; 计算小端半径

G00 X[#1/2+#2] Y0; 快速定位到刀尖与大端外轮廓刀具半径值处
Z-#3; 快速下降到总深度

#26=#3; 将总深度作为初始值赋给背吃刀量#26

#8=0; 水平旋转角度赋初始值为 0 度

WHILE[#26GE0] DO1; 循环 1 设定条件: 背吃刀量#26 不小于 0

G68 X0 Y0 R#8; 坐标系绕 X0 Y0 旋转#8 角度

#24=#5+#26/#4/2-#11; 计算相应水平 X 坐标值

#14=#24+#18; 计算 1/4 圆弧切入起始点 X 坐标

#15=#13-#26; 计算深度相应坐标值

G00 G41 D1 X#14 Y#18 Z#15; 半径补偿定位到 1/4 圆弧切入起始点

G03 X#24 Y0 R#18 F#7; 圆弧切入

G02 I-#24; 顺时针整圆圆弧切削

G03 X#14 Y-#18 R#18; 1/4 圆弧切出

G00 G40 Y0; 取消半径补偿

#26=#26-#6; 计算下一轮背吃刀量

#8=#8+#17; 水平旋转角度累加一个角度递增量#17

G69; 坐标系旋转取消

END1; 循环 1 结束

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O124;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6124 A70 B16 C15 I5 D1000 K0.5 R10 T3 Q10; 调用宏程序 O6124, 相应变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(3) 球头铣刀

1) 分析: 如图 6-10 所示, 这里用的刀具是球头铣刀, 刀具参考点是球心, 没有用半径补偿功能, 刀具参考点坐标计算与上两例有区别, 其他相同。加工路

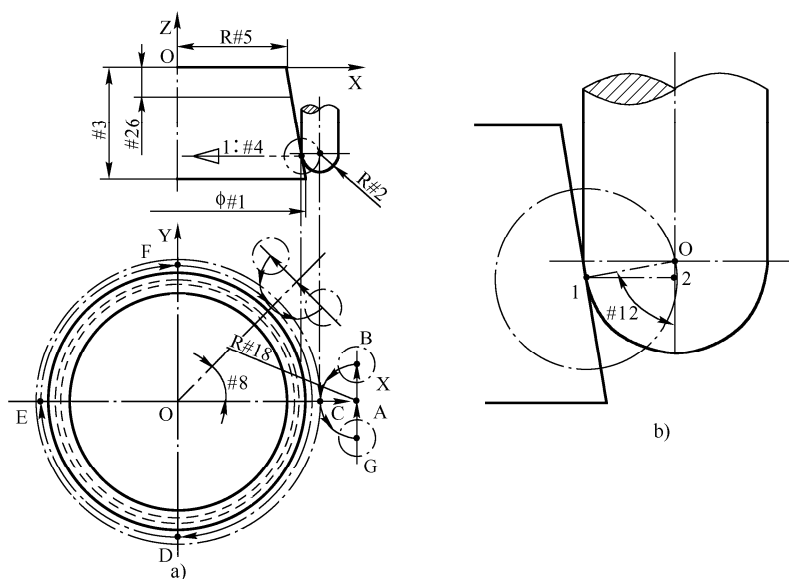


图 6-10 外圆锥体精加工 (3)

a) 整体示意图 b) 局部放大图

线轨迹: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow C \rightarrow G \rightarrow A$ 。

2) 参数及说明:

#1=[A]; 锥体大端直径

#2=[B]; 球头铣刀半径

#3=[C]; 锥体高度

#4=[I]; 锥度 1 : #4

#6=[K]; 背吃刀量增量

#7=[D]; 进给量

#17=[Q]; 角度递增量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6125; 子程序号

#12=ATAN[#4]/[2]; 计算圆锥大端锥角

#11=#20*[SIN[#12]-1]; 计算刀具中心点与切削点 X 轴坐标值差值

#13=#20*[COS[#12]-1]; 计算刀具中心点与切削点 Z 轴坐标值差值

#5=[#1-#3/#4]/2; 计算小端半径

G00 X[#1/2+#18+#2] Y0; 快速定位到刀尖与大端外轮廓距离圆弧半径值处
Z-#3; 快速下降到总深度

#26=#3; 将总深度作为初始值赋给背吃刀量#26

#8=0; 水平旋转角度赋初始值为 0 度

WHILE[#26GE0] DO1; 循环 1 设定条件: 背吃刀量#26 不小于 0

G68 X0 Y0 R#8; 坐标系绕 X0 Y0 旋转#8 角度

#24=#5+#26/#4/2+#11; 计算相应水平 X 坐标值

#14=#24+#18; 计算 1/4 圆弧切入起始点 X 坐标

#15=#13-#26; 计算深度相应坐标值

G00 X#14 Y#18 Z#15; 快速定位到圆弧切入起始点

G03 X#24 Y0 R#18 F#7; 圆弧切入

G02 I-#24; 顺时针整圆圆弧切削

G03 X#14 Y-#18 R#18; 1/4 圆弧切出

G00 X[#1/2+#18+#2] Y0; 返回切削准备点

#26=#26-#6; 计算下一轮背吃刀量

#8=#8+#17; 水平旋转角度累加一个角度递增量#17

G69; 坐标系旋转取消

END1; 循环 1 结束

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O125;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6125 A70 B8 C15 I5 D1000 K0.5 R10 Q10; 调用宏程序 O6125,
相应变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

3. 内圆锥体粗加工

(1) 立铣刀 (由上而下)

1) 分析: 如图 6-11 所示, 参照外圆锥体编程思路, 只是加工起始点在孔内, 整圆铣削。为了保证顺铣, 采用的是顺时针铣削路线, 这跟外圆锥体正好相反, 其他借鉴外圆锥体方法。加工路线轨迹: A→B→C→D→E→B→A。

2) 参数及说明:

#1=[A]; 锥体小端直径

#2=[B]; 刀具直径

#3=[C]; 锥体高度

#4=[I]; 锥度 1: #4

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

3) 宏程序如下:

%

O6126; 子程序号

#5=[#1+#3/#4]/2; 计算圆锥小端半径

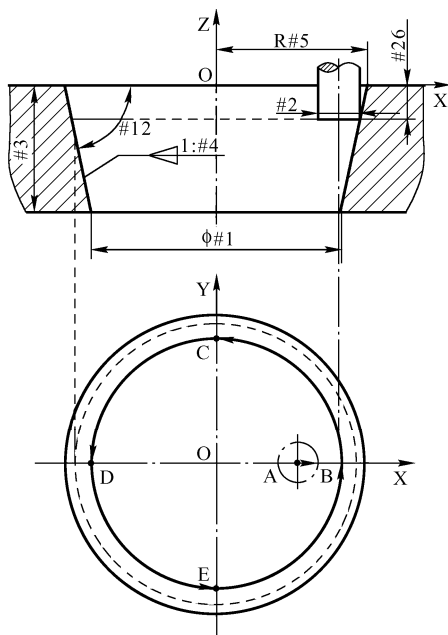


图 6-11 内圆锥体粗加工 (1)

```

G00 X[#1-#2]/2-2] Y0; 快速定位到距内孔壁 X 轴方向中心 2mm 处
Z5; 快速下降至 Z=5 处
#26=#6; 给深度赋初始值一个背吃刀量
WHILE[#26LE#3] DO1; 循环 1 设定条件: 背吃刀量#26 不大于总深度
#24=#6-#2/2-#26/#4/2; 计算相应水平 X 坐标值
Z-#26; 快速下降到背吃刀量#26
G01 X#24 F#7; 直线切削到该深度对应的 X 坐标值#24
G02 I-#24; 逆时针整圆圆弧切削
G00 Z[2-#26]; 抬刀 2mm
X[#1-#2]/2-2] Y0; 返回刚才起始点水平位置
#26=#26+#6; 计算下一轮背吃刀量
END1; 循环 1 结束
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
%
```

4) 主程序如下:

```

%
O126;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S400;
G65 P6126 A60 B16 C15 I5 D300 K1.5; 调用宏程序 O6126, 相应
                                         变量赋初始值

G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

(2) 圆鼻刀

1) 分析: 如图 6-12 所示, 参照上例, 只是刀具不同, 用的是圆鼻刀, 坐标计算有点区别, 其他相同。加工路线轨迹: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow B \rightarrow A$ 。

2) 参数及说明:

#1=[A]; 锥体小端直径

#2=[B]; 刀具直径

#3=[C]; 锥体高度

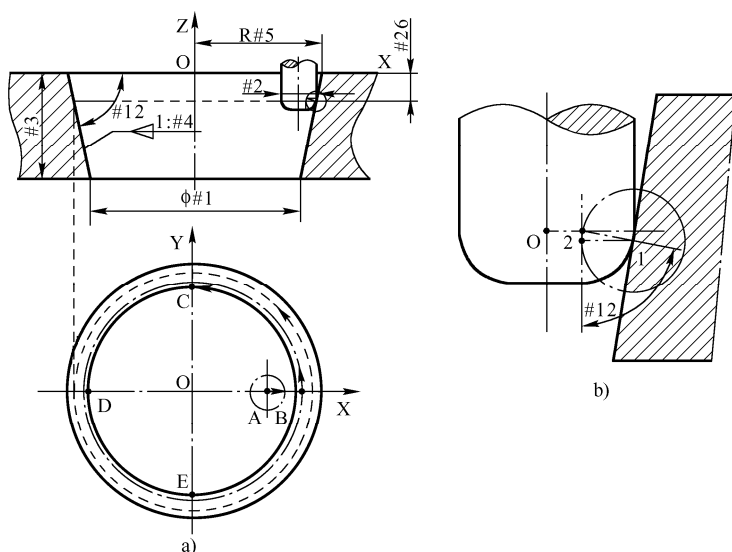


图 6-12 内圆锥体粗加工 (2)

a) 整体示意图 b) 局部放大图

#4=[I]; 锥度 1 : #4

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

#20=[T]; 刀尖圆弧半径

3) 宏程序如下:

%

O6127; 子程序号

#12=ATAN[#4]/[2]; 计算圆锥大端锥角

#11=#20*[SIN[#12]-1]; 计算刀具中心点与切削点 X 轴坐标值差值

#13=#20*[COS[#12]-1]; 计算刀具中心点与切削点 Z 轴坐标值差值

#5=[#1+#3/#4]/2; 计算圆锥大端半径

G00 X[[#1-#2]/2-2] Y0; 快速定位到刀尖与小端外轮廓 2mm 处
Z5; 快速下降到 Z=5

#26=#6; 将深度增量#6 作为初始值赋给背吃刀量#26

WHILE[#26LE#3] DO1; 循环 1 设定条件: 背吃刀量#26 不大于总深度

#24=#6-#2/2-#26/#4/2+#11; 计算相应水平 X 坐标值

Z[#13-#26]; 计算相应深度 Z 坐标值

G01 X#24 F#7; 水平直线切削到整圆切削起始点

G03 I-#24; 时针整圆圆弧切削

```

G00 Z[2-#26]; 抬刀 2mm
X[#1-#2]/2-2] Y0; 返回起始点
#26=#26+#6; 计算下一轮背吃刀量
END1; 循环 1 结束
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
%
```

4) 主程序如下:

```
%
```

```
O127;
```

```
G54 G90 G00 G43 H1 Z50;
```

```
X0 Y0;
```

```
M03 S400;
```

```
G65 P6127 A60 B16 C15 I5 D300 K1.5 T3; 调用宏程序 O6127,
相应变量赋初始值
```

```
G00 G49 Z100;
```

```
X0 Y150;
```

```
M05;
```

```
M30;
```

```
%
```

4. 内圆锥体精加工

(1) 立铣刀 (由下而上)

1) 分析: 如图 6-13 所示, 由下而上, 逐层逆时针整圆铣削, 参照上面实例。加工路线轨迹: A→B→C→E→F→G→C→D→A。

2) 参数及说明:

#1=[A]; 锥体 v 端直径

#2=[B]; 刀具直径

#3=[C]; 锥体高度

#4=[I]; 锥度 1: #4

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

#17=[Q]; 角度递增量

#18=[R]; 1/4 圆弧切入切出半径

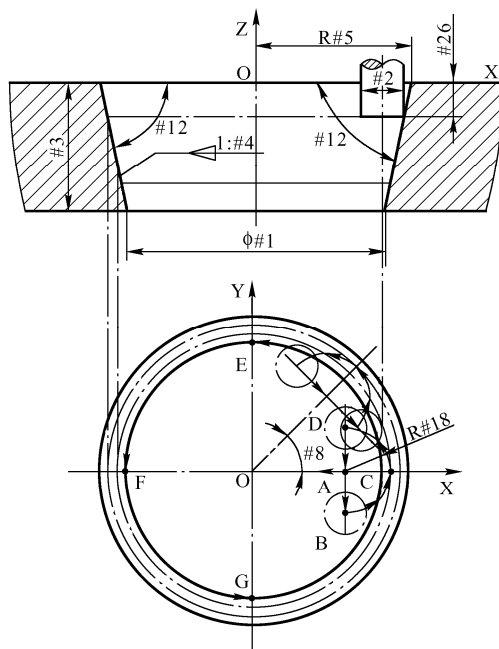


图 6-13 内圆锥体精加工 (1)

3) 宏程序如下:

%

O6128; 子程序号

#5=[#1+#3/#4]/2; 计算圆锥大端半径

G00 X0 Y0; 快速定位到 X0 Y0

Z-#3; 快速下降到总深度处

#26=#3; 将总深度#3 作为初始值赋给背吃刀量#26

#8=0; 水平旋转角度赋初始值为 0 度

WHILE[#26GE0] DO1; 循环 1 设定条件: 背吃刀量#26 不小于 0

G68 X0 Y0 R#8; 坐标系旋转#8

#24=#6-#26/#4/2; 计算整圆切削起始点 X 坐标值

#14=#24-#18; 计算 1/4 圆弧切入起始点 X 坐标值

G00 G41 D1 X#14 Y-#18; 半径补偿到圆弧切入起始点

G03 X#24 Y0 Z-#26 R#18 F#7; 圆弧切入到整圆切削起始点

G03 I-#24; 逆时针整圆圆弧切削

G03 X#14 Y#18 R#18; 圆弧切出

G00 G40 X0 Y0; 快速水平返回到 X0 Y0

#26=#26-#6; 计算下一轮背吃刀量

#8=#8+#17; 水平旋转角度累加一个角度递增量#17

G69; 坐标系旋转取消

END1; 循环 1 结束

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O128;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6128 A60 B16 C15 I5 D1000 K0.5 Q10 R10; 调用宏程序 O6128, 相应
变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 圆鼻刀

1) 分析: 如图 6-14 所示, 参照上例, 刀具参考点坐标计算有所区别, 其他相同。加工路线轨迹: $A \rightarrow B \rightarrow C \rightarrow E \rightarrow F \rightarrow G \rightarrow C \rightarrow D \rightarrow A$ 。

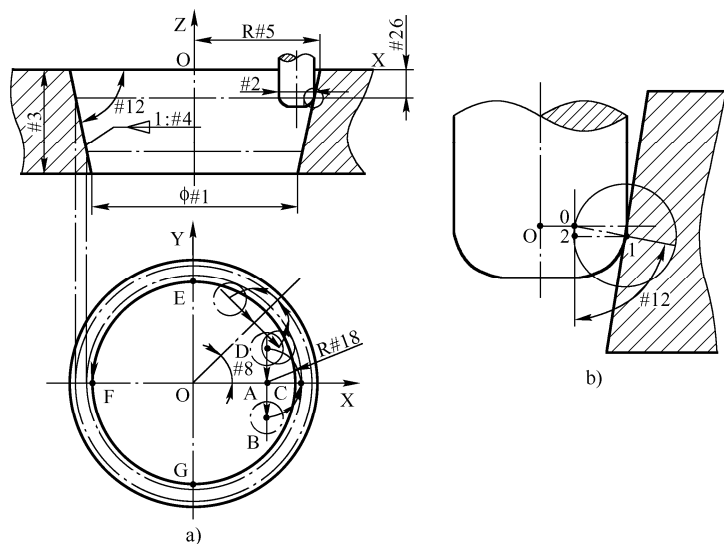


图 6-14 内圆锥体精加工 (2)

a) 整体分布图 b) 局部放大图

2) 参数及说明:

#1=[A]; 锥体小端直径

#2=[B]; 刀具直径

#3=[C]; 锥体高度

#4=[I]; 锥度 1 : #4

#6=[K]; 背吃刀量递增量

#7=[D]; 进给量

#17=[Q]; 角度递增量

#18=[R]; 1/4 圆弧切入切出半径

#20=[T]; 刀尖圆弧半径

3) 宏程序如下:

%

O6129; 子程序号

#12=ATAN[#4]/[2]; 计算圆锥大端锥角

#11=#20*[SIN[#12]-1]; 计算刀具中心与切削点 X 轴坐标差值
#13=#20*[COS[#12]-1]; 计算刀具中心与切削点 Z 轴坐标差值
#5=[#1+#3/#4]/2; 计算圆锥大端半径
G00 X0 Y0; 快速定位到 X0 Y0
Z-#3; 快速下降到总深度
#26=#3; 将总深度#3 作为初始值赋给背吃刀量#26
#8=0; 水平旋转角度赋初始值 0
WHILE[#26GE0] DO1; 循环 1 设定条件: 背吃刀量#26 不小于 0
#24=#6-#26/#4/2+#11; 计算整圆切削起始点 X 坐标值
G68 X0 Y0 R#8; 坐标系旋转#8
#14=#24-#18; 计算 1/4 圆弧切入切出起始点 X 轴坐标值
#15=#13-#26; 计算整圆背吃刀量值
G00 G41 D1 X#14 Y-#18 Z#15; 半径补偿到圆弧切入起始点
G03 X#24 Y0 R#18 F#7; 圆弧切入
G03 I-#24; 逆时针整圆圆弧切削
G03 X#14 Y#18 R#18; 圆弧切出
G00 G40 X0 Y0; 取消半径至 X0 Y0
#26=#26-#6; 计算下一轮背吃刀量
#8=#8+#17; 水平旋转角度累加一个角度递增量#17
G69; 坐标系旋转取消
END1; 循环 1 结束
G00 Z50; 快速上升至 Z=50
M99; 子程序结束

%

4) 主程序如下:

%

O129;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6129 A60 B16 C15 I5 D1000 K0.5 R10 T3 Q10; 调用宏程序 O6129, 相应变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(3) 球头铣刀

1) 分析: 如图 6-15 所示, 此例刀具用的是球头铣刀, 不用半径补偿, 球作为刀具参考点, 刀具坐标计算要注意这点, 其他与上例相同。加工路线轨迹: A→B→C→E→F→G→C→D→A。

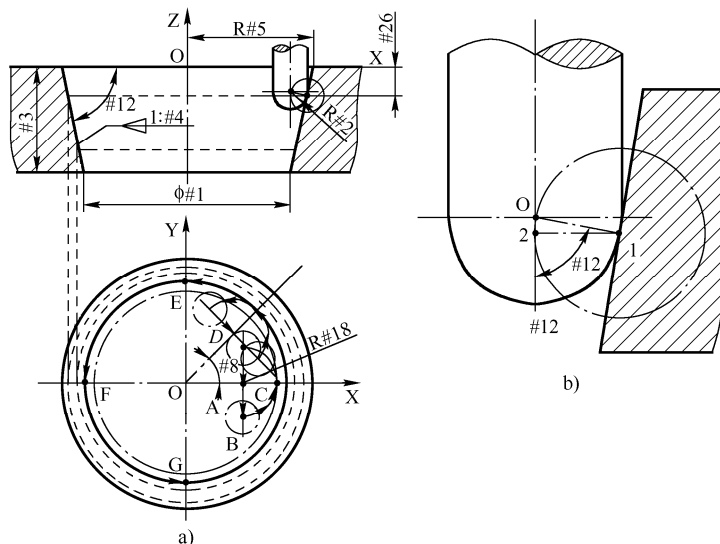


图 6-15 内圆锥体精加工 (3)

a) 整体示意图 b) 局部放大图

2) 参数及说明:

#1=[A]; 锥体小端直径

#2=[B]; 球头铣刀半径

#3=[C]; 锥体高度

#4=[I]; 锥度 1 : #4

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

#17=[Q]; 角度递增量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6120; 子程序号

#12=ATAN[#4]/[2]; 计算圆锥大端锥角

```
#11=#2*SIN[#12]; 计算球头铣刀球心与切削点 X 轴坐标值差值
#13=#2*COS[#12]; 计算球头铣刀球心与切削点 Z 轴坐标值差值
#5=[#1+#3/#4]/2; 计算圆锥大端半径
G00 X0 Y0; 快速定位到 X0 Y0
Z-#3; 快速下降到总深度
#26=#3; 将总深度#3 作为初始值赋给背吃刀量#26
#8=0; 水平旋转角度赋初始值为 0
WHILE[#26GE0] DO1; 循环 1 设定条件: 背吃刀量#26 不小于 0
#24=#6-#26/#4/2-#11; 计算整圆切削起始点 X 轴坐标值
G68 X0 Y0 R#8; 坐标系旋转#8
#14=#24-#18; 计算 1/4 圆弧切入切出起始点 X 轴坐标值
#15=#13-#26; 计算整圆切削起始点 Z 轴坐标值
G01 X#14 Y-#18 Z#15 F#7; 直线切削到 1/4 圆弧切入起始点
G03 X#24 Y0 R#18; 圆弧切入
G03 I-#24; 整圆圆弧切削
G03 X#14 Y#18 R#18; 圆弧切出
G00 X0 Y0; 水平返回到 X0 Y0
#26=#26-#6; 计算下一轮背吃刀量
#8=#8+#17; 水平旋转角度累加一个角度递增量#17
G69; 坐标系旋转取消
END1; 循环 1 结束
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
```

%

4) 主程序如下:

%

O120;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6120 A60 B8 C15 I5 D1000 K0.5 Q10; 调用宏程序 O6120,
相应变量赋初始值

G00 G49 Z100;

X0 Y150

M05;

M30;

%

6.1.3 实例三棱柱加工

1. 外棱柱加工（立铣刀、由上而下）

1) 分析：如图 6-16 所示，首先运用坐标系旋转指令 G68 将坐标系调整到棱柱一棱面垂直于 X 轴，且对称于 X 轴，其次利用极坐标指令简化坐标计算，棱柱的五个顶点坐标由极半径（ $\#1/2$ ）和极角（ $\#3 + [180/\#2] * 2 * \#13$ ）确定，其中 $\#1$ ：外接圆直径， $\#2$ ：棱柱边数， $\#13$ ：从 X 轴正向逆时针棱柱顶点序号（从 0 开始计数）。

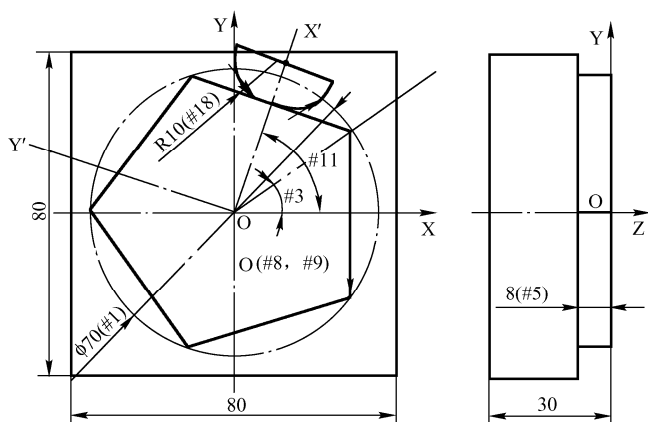


图 6-16 外棱柱加工

2) 参数及说明：

$\#1=[A]$ ；外接圆直径

$\#2=[B]$ ；正多边形边数 N

$\#3=[C]$ ；第一顶点与 X 轴正向夹角（最小正角）

$\#5=[J]$ ；凸台高度

$\#6=[K]$ ；高度递增量

$\#7=[D]$ ；进给量

$\#18=[R]$ ；1/4 圆弧切入切出半径

3) 宏程序如下：

%

O6131；子程序号

$\#10=\text{FIX}[\#5/\#6]$ ；计算深度切削次数

$\#16=180/\#2$ ；计算棱柱棱角半角值

$\#11=\#3+\#16$ ；计算坐标系旋转角度值

```
G68 X0 Y0 R#11; 坐标系旋转#11
#20=0.5*#1*COS[#16]+#18; 计算 1/4 圆弧切入起始点 X 轴坐标值
G00 X#20 Y0; 快速定位到建立半径补偿起始点
Z5; 快速下降到 Z=5 处
#12=0; 给深度切削次数计数器赋初始值为 0
WHILE[#12LE#10] DO2; 循环 2 设定条件: 计数器#12 值不大于深度切削
    次数#10
G41 G00 D1 Y[#18+2]; 半径补偿到圆弧切入起始点
#26=#6*[#12+1]; 每轮背吃刀量值计算
IF[#26GT#5] THEN#26=#5; 如果背吃刀量值大于总深度, 则限定它最大为
    总深度
G03 X[#20-#18] Y2 Z-#26 R#18 F#7; 螺旋切入
#13=0; 棱边数计数器值赋初始值为 0
#14=#16; 给顶点极角赋初始值#16
G16; 建立极坐标系
WHILE[#13LT#2] DO1; 循环 1 设定条件: 计数器#13 值不大于棱边数#2
#14=#16+#16*2*#13; 计算顶点对应的极角
G01 X[#1/2] Y-#14; 直线切削到对应顶点
#13=#13+1; 棱边数计数器值累加 1
END1; 循环 1 结束
G15; 极坐标系取消
G01 Y-2; 直线切削到完整棱柱
G03 X#20 Y-[#18+2] Z[2-#26] R#18; 螺旋切出, 刀具抬高 2mm
G00 G40 Y0; 取消半径补偿
#12=#12+1; 深度计数器值累加 1
END2; 循环 2 结束
G69; 坐标系旋转取消
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
%
4) 主程序如下:
%
O131;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
```

```

M03 S400;
G65 P6131 A70 B5 C25 I5 J8 K1.5 D300 R10; 调用宏程序 O6131,
                                             赋相关变量初始值

G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

2. 带圆角内棱柱加工（立铣刀、由上而下）

1) 分析：如图 6-17b 所示，C、E 为圆角的两个切点，A、F 分别为相邻两棱边的中点，由此可知此四点也为垂足，即 $\triangle ABO$ 、 $\triangle BCD$ 、 $\triangle BED$ 、 $\triangle BOF$ 都是直角三角形。现设定 $OA=\#20$ ， $AB=\#21$ ， $BC=\#22$ ， $AC=\#23$ ， $OC=\#9$ ； $\angle AOC=\#8$ ， $\angle BOC=\#17$ ， $\angle AOB=\#16$ 。

$OB=\#1/2$ ， $OA=\#20=OB*\cos\angle AOB=\#1/2*\cos[\#16]$ ， $AB=\#21=OB*\sin\angle AOB=\#1/2*\sin[\#16]$ ， $BC=\#22=DC*\tan\angle BDC=\#4*\tan[\#16]$ ， $AC=\#23=AB-BC=\#21-\#22$ ， $\angle AOC=\#8=\text{ATAN}[AC/AO]=\text{ATAN}[\#23]/[\#20]$ ， $OC=\#9=\sqrt{AC^2+AO^2}=\text{SQRT}[\#20*\#20+\#23*\#23]$ 。

C 的 X、Y 坐标值分别为：

$$\#14=\#9*\cos[\#8]$$

$$\#15=\#9*\sin[\#8]$$

E 的 X、Y 坐标值分别为：

$$\#24=\#9*\cos[\#8+2*\#17]$$

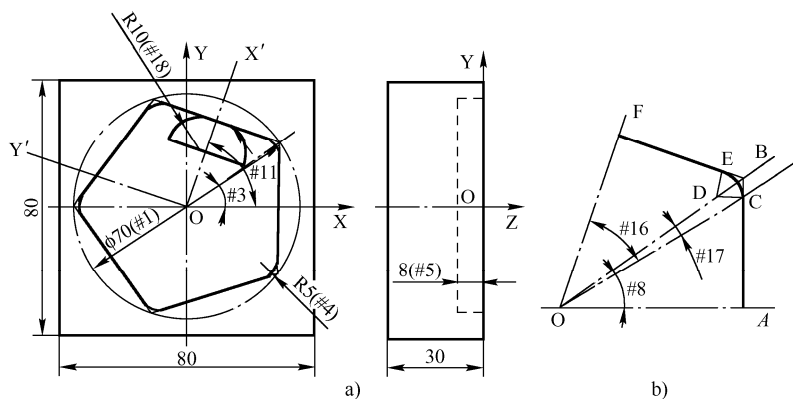


图 6-17 带圆角内棱柱加工

a) 零件图 b) 局部放大图

```
#25=#9*SIN[#8+2*#17]
```

假设棱边计数器为#13, 赋初始值为 0, 则对应 C、E 点后面的点的 X、Y 坐标值分别为:

```
#14=#9*COS[#8+#16*#13]
```

```
#15=#9*SIN[#8+#16*#13]
```

```
#24=#9*COS[#8+2*#17+#16*#13]
```

```
#25=#9*SIN[#8+2*#17+#16*#13]
```

所以切削加工从螺旋切入到 A 点, 直线插补到 C 点, 再逆时针圆弧插补到 E 点; 然后直线、圆弧插补到相对 C、E 两点的下一轮对应点, 进而完成整个轮廓的加工。

2) 参数及说明:

#1=[A]; 外接圆直径

#2=[B]; 内棱柱边数 N

#3=[C]; 第一顶点与 X 轴正向夹角 (最小正角)

#4=[I]; 正多边形顶点倒圆半径

#5=[J]; 内棱柱高度

#6=[K]; 深度递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6132; 子程序号

#10=FIX[#5/#6]; 计算深度切削次数

#16=180/#2; 计算棱柱棱角半角值

#11=#3+#16; 计算坐标系旋转角度值

G68 X0 Y0 R#11; 坐标系旋转#11

#20=0.5*#1*COS[#16]; 计算棱柱底面中心到棱面的距离 (OA)

#21=0.5*#1*SIN[#16]; 计算棱柱的棱面宽度半值 (AB)

#22=#4*TAN[#16]; 计算在棱柱底面上倒圆切点到棱面未倒圆前棱边的距离 (BC)

#23=#21-#22; 计算倒圆后棱面两切线间距离的半值 (AC)

#9=SQRT[#20*#20+#23*#23]; 计算棱柱底面上中心到切点的距离 (OC)

#8=ATAN[#23]/[#20]; 计算 $\angle AOC$ 角度值

#17=#16-#8; 计算棱柱棱角半值与#8 的差值

G00 X0 Y0; 快速定位到 (0, 0)

```

Z5; 快速下降到 Z=5 处
#12=0; 给深度切削次数计数器赋初始值为 0
WHILE[#12LE#10] DO2; 循环 2 设定条件: 计数器#12 值不大于深度切削
                        次数#10
G41 G00 D1 X[#20-#18] Y-[#18+2]; 半径补偿到圆弧切入起始点, 增
                        加切入切出重叠量 4mm
#26=#6*[#12+1]; 每轮背吃刀量值计算
IF[#26GT#5] THEN#26=#5; 如果背吃刀量值大于总深度, 则限定它最大为
                        总深度
G03 X#20 Y-2 Z-#26 R#18 F#7; 螺旋切入
#13=0; 棱边数计数器值赋初始值为 0
WHILE[#13LT#2] DO1; 循环 1 设定条件: 计数器#13 值小于棱边数#2
#30=#16*#13*2;
#14=#9*COS[#8+#30]; 计算棱面上切点的 X 轴坐标
#15=#9*SIN[#8+#30]; 计算棱面上切点的 Y 轴坐标
#24=#9*COS[#8+2*#17+#30]; 计算下一棱面上切点的 X 轴坐标
#25=#9*SIN[#8+2*#17+#30]; 计算下一棱面上切点的 Y 轴坐标
G01 X#14 Y#15; 直线切削到第一个切点 (#14, #15)
G03 X#24 Y#25 R#4; 逆圆切削到第二个切点 (#24, #25)
#13=#13+1; 棱边数计数器值累加 1
END1; 循环 1 结束
G01 Y2; 直线切削到完整棱柱, 保证重叠量 4mm
G03 X[#20-#18] Y[#18+2] Z[2-#26] R#18; 螺旋切出, 刀具抬高 2mm
G00 G40 X0 Y0; 取消半径补偿
#12=#12+1; 深度计数器值累加 1
END2; 循环 2 结束
G69; 坐标系旋转取消
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
%
4) 主程序如下:
%
O132;
G54 G90 G00 G43 H1 Z50;
X0 Y0;

```



```

M03 S400;
G65 P6132 A70 B5 C25 I10 J8 K1.5 D300 R10; 调用宏程序 O6132,
                                             给相关变量赋初始值

G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

3. 带圆角外棱柱加工（立铣刀、由上而下）

1) 分析：如图 6-18 所示，它与图 6-17 的区别仅仅是内外形，只需将圆弧切入切出环节由内调整到外，切入切出起始点 X 轴坐标由[#20-#18]改为[#20+#18]，Y 轴坐标正负对调，轮廓铣削是顺时针方向，其他一样。

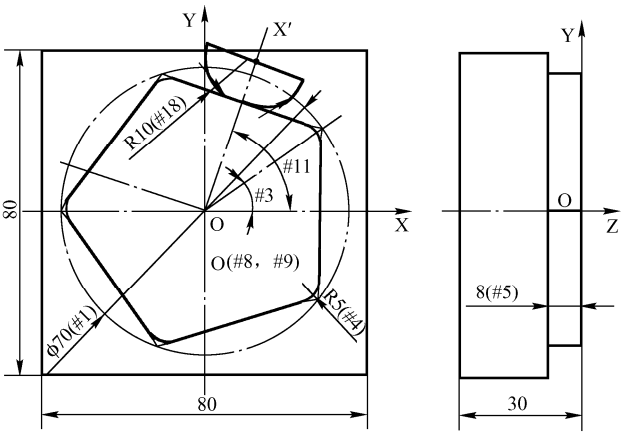


图 6-18 带圆角外棱柱加工

2) 参数及说明：

- #1=[A]；外接圆直径
- #2=[B]；外棱柱边数 N
- #3=[C]；第一顶点与 X 轴正向夹角（最小正角）
- #4=[I]；顶点倒圆半径
- #5=[J]；凸台高度
- #6=[K]；高度递增量
- #7=[D]；进给量
- #18=[R]；1/4 圆弧切入切出半径

3) 宏程序如下：

%

O6133; 子程序号

#10=FIX[#5/#6]; 计算深度切削次数

#16=180/#2; 计算棱柱棱角半角值

#11=#3+#16; 计算坐标系旋转角度值

G68 X0 Y0 R#11; 坐标系旋转#11

#20=0.5*#1*COS[#16]; 计算棱柱底面中心到棱面的距离

#21=0.5*#1*SIN[#16]; 计算棱柱的棱面宽度半值

#22=#4*TAN[#16]; 计算在棱柱底面上倒圆切点到棱面未倒圆前棱边的距离

#23=#21-#22; 计算倒圆后棱面两切线间距离的半值

#9=SQRT[#20*#20+#23*#23]; 计算棱柱底面上中心到切点的距离

#8=TAN[#23/#20]; 计算切点与底面中心连线跟中心到棱边高之间的夹角

#17=#16-#8; 计算棱柱棱角半值与#8的差值

G00 X[#20+#18] Y0; 快速定位到加工准备位置点(#20+#18, 0)

Z5; 快速下降到Z=5处

#12=0; 给深度切削次数计数器赋初始值为0

WHILE[#12LE#10] DO2; 循环2 设定条件: 计数器#12值不大于深度切削次数#10

G41 G00 D1 Y[#18+2]; 半径补偿到圆弧切入起始点, 增加切入切出重叠量4mm

#26=#6*[#12+1]; 每轮背吃刀量值计算

IF[#26GT#5] THEN#26=#5; 如果背吃刀量值大于总深度, 则限定它最大为总深度

G03 X#20 Y2 Z-#26 R#18 F#7; 螺旋切入

#13=0; 棱边数计数器值赋初始值为0

WHILE[#13LT#2] DO1; 循环1 设定条件: 计数器#13值小于棱边数#2

#30=#16*#13*2;

#14=#9*COS[-#8-#30]; 计算棱面上切点的X轴坐标

#15=#9*SIN[-#8-#30]; 计算棱面上切点的Y轴坐标

#24=#9*COS[-#8-2*#17-#30]; 计算下一棱面上切点的X轴坐标

#25=#9*SIN[-#8-2*#17-#30]; 计算下一棱面上切点的Y轴坐标

G01 X#14 Y#15; 直线切削到第一个切点(#14, #15)

G02 X#24 Y#25 R#4; 顺圆切削到第二个切点(#24, #25)

#13=#13+1; 棱边数计数器值累加1

END1; 循环1结束

```
G01 Y-2; 直线切削到完整棱柱, 保证重叠量 4mm
G03 X[#20+#18] Y-[#18+2] Z[2-#26] R#18; 螺旋切出, 刀具抬高 2mm
G00 G40 Y0; 取消半径补偿
#12=#12+1; 深度计数器值累加 1
END2; 循环 2 结束
G69; 坐标系旋转取消
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
%
4) 主程序如下:
%
O133;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S400;
G65 P6133 A70 B5 C25 J8 K1.5 D300 R10; 调用宏程序 O6133,
                                         给相关变量赋初始值
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.1.4 实例四棱锥加工

1. 外棱锥台粗加工

1) 分析: 如图 6-19 所示, 用立铣刀对底面外接圆直径为#1, 顶面外接圆直径为#2, 高#3, 棱边数#4 的外棱锥台进行粗加工。建立如图 6-19 所示坐标系, 以顶面中心为坐标系原点, 向上为 Z 轴正向。

首先利用坐标系旋转指令将顶面一边对称垂直于 X 轴正向, 见图 6-19 中 X'OY'坐标系。以#6 为深度变量, 从顶面每次下降一个#6 深度变量, 计算此时该层棱锥截面正多边形相关尺寸, 螺旋切入、半径补偿、顺时针走刀切削此多边形、螺旋切出; 一层一层如此反复, 直至切削到底面。

2) 变量及说明:

#1=[A]; 大端外接圆直径

#2=[B]; 小端外接圆直径

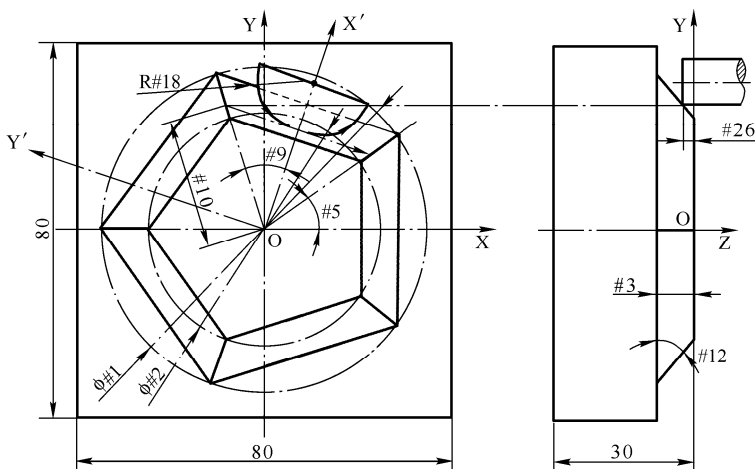


图 6-19 外棱锥台粗加工

#3=[C]; 棱锥高

#4=[I]; 棱锥棱边数

#5=[J]; 第一顶点与 X 轴正向夹角

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

06141; 子程序号

#12=ATAN[2*#3]/[#1-#2]; 计算锥台棱线锥角

#9=180/#4; 计算棱锥底面正多边形中心角半角值

G68 X0 Y0 R[#5+#9]; 坐标系旋转, 保证顶面一边垂直于 X 轴并对称于 X 轴

#26=#6; 给背吃刀量#26 赋初始值一个切削增量#6

WHILE[#26LE#3] DO2: 循环 2 设定条件: 背吃刀量#26 不大于深度#3

#10=#2/2+#26/TAN[#12]; 计算任意切削层外接圆半径值

#11=#10*COS[#9]; 计算任意切削层棱边中心点 X 轴坐标值

G00 X[#11+#18] Y0; 快速定位到切削准备点

Z[3-#26]: 快速下降到离背吃刀量 3mm 处

G41 D1 Y[#18+2]; 建立半径补偿

G03 X#11 Y2 Z-#26 R#18 F#7; 螺旋切入

```

#20=-#9; 给棱边任意顶点与 X 轴夹角赋初始值-#9
WHILE[#20GT-360] DO1; 循环 1 设定条件: 底面顶点与 X 轴正向夹角大于-360°
#24=#10*COS[#20]; 计算任意切削层棱边顶点的 X 轴坐标
#25=#10*SIN[#20]; 计算任意切削层棱边顶点的 Y 轴坐标
G01 X#24 Y#25; 直线切削到棱边顶点
#20=#20-2*#9; 夹角#20 递减一个棱锥底面正多边形中心角
END1; 循环 1 结束
G01 Y-2; 完成一层轮廓切削
G03 X[#11+#18] Y-[#18+2] Z[3-#26] R#18; 螺旋切出, 抬刀 3mm
G00 G40 Y0; 取消半径补偿
#26=#26+#6; 背吃刀量累加一个深度增量#6
END2; 循环 2 结束
G69; 坐标系旋转取消
G00 Z50; 快速上升至 Z=50
M99; 子程序结束

```

%

4) 主程序如下:

%

O141;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S400;

G65 P6141 A75 B60 C8 I5 J15 K1.5 D300 R15; 调用宏程序 O6141,
赋相关变量初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 外棱锥台精加工

(1) 立铣刀

1) 分析: 如图 6-19 所示, 用立铣刀对底面外接圆直径为#1, 顶面外接圆直径为#2, 高#3, 棱边数#4 的外棱锥台进行精加工。建立如图 6-19 所示坐标系, 以顶面中心为坐标系原点, 向上为 Z 轴正向。

与上例粗加工相似，只是开始从下向上，且每次深度增量#6 变小。

2) 变量及说明：

#1=[A]；大端外接圆直径

#2=[B]；小端外接圆直径

#3=[C]；棱锥高

#4=[I]；棱锥棱边数

#5=[J]；第一顶点与 X 轴正向夹角

#6=[K]；每次背吃刀量增量

#7=[D]；进给量

#18=[R]；1/4 圆弧切入切出半径

3) 宏程序如下：

%

O6142；子程序号

#12=ATAN[2*#3]/[#1-#2]；计算锥台棱线锥角

#9=180/#4；计算棱锥底面正多边形中心角半角值

G68 X0 Y0 R[#5+#9]；坐标系旋转，保证顶面一边垂直于 X 轴并对称于 X 轴

#26=#3；给背吃刀量#26 赋初始值一个切削增量#3

WHILE[#26GE0] DO2；循环 2 设定条件：背吃刀量#26 不小于深度 0

#10=#2/2+#26/TAN[#12]；计算任意切削层外接圆半径值

#11=#10*COS[#9]；计算任意切削层棱边中心点 X 轴坐标值

G00 X[#11+#18] Y0；快速定位到切削准备点

Z[3-#26]；快速下降到离背吃刀量 3mm 处

G41 D1 Y[#18+2]；建立半径补偿

G03 X#11 Y2 Z-#26 R#18 F#7；螺旋切入

#20=-#9；给棱边任意顶点与 X 轴夹角赋初始值-#9

WHILE[#20GT-360] DO1；循环 1 设定条件：底面顶点与 X 轴正向夹角大于-360°

#24=#10*COS[#20]；计算任意切削层棱边顶点的 X 轴坐标

#25=#10*SIN[#20]；计算任意切削层棱边顶点的 Y 轴坐标

G01 X#24 Y#25；直线切削到棱边顶点

#20=#20-2*#9；夹角#20 递减一个棱锥底面正多边形中心角

END1；循环 1 结束

G01 Y-2；完成一层轮廓切削

G03 X[#11+#18] Y-[#18+2] Z[3-#26] R#18；螺旋切出，抬刀 3mm

G00 G40 Y0; 取消半径补偿

#26=#26-#6; 背吃刀量递减一个深度增量#6

END2; 循环 2 结束

G69; 坐标系旋转取消

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O142;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S400;

G65 P6142 A75 B60 C8 I5 J15 K0.4 D1000 R15; 调用宏程序 O6142,
赋相关变量初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 球头铣刀

1) 分析: 如图 6-20 所示, 用半径#8 球头铣刀对底面外接圆直径为#1, 顶面外接圆直径为#2, 高#3, 棱边数#4 的外棱锥台进行精加工。建立如图 6-20 所示坐标系, 以顶面中心为坐标系原点, 向上为 Z 轴正向。

首先利用坐标系旋转指令将顶面一边对称垂直于 X 轴正向, 见图 6-20 中 X'OY'坐标系。以#6 为深度变量, 从底面依次上升一个#6 深度变量, 计算此时该层棱锥截面正多边形相关尺寸, 螺旋切入、顺时针走刀切削此多边形、螺旋切出; 一层一层如此反复, 直至切削到顶面。

2) 变量及说明:

#1=[A]; 大端外接圆直径

#2=[B]; 小端外接圆直径

#3=[C]; 棱锥高

#4=[I]; 棱锥棱边数

#5=[J]; 第一顶点与 X 轴正向夹角

#6=[K]; 每次背吃刀量增量

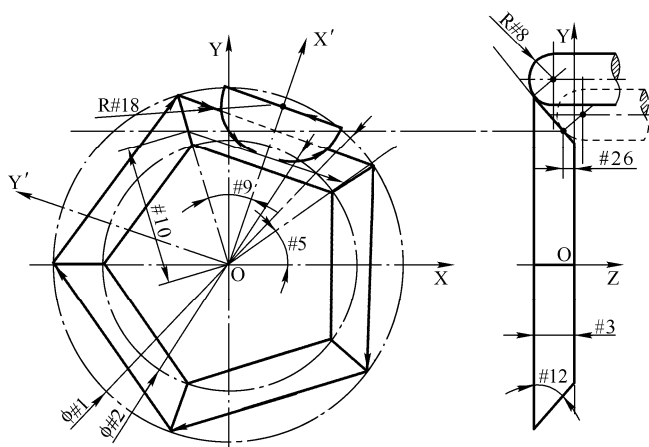


图 6-20 外棱锥台精加工

#7=[D]; 进给量

#8=[E]; 球头铣刀半径

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6143; 子程序号

#12=ATAN[2*#3]/[#1-#2]; 计算锥台棱线锥角

#9=180/#4; 计算棱锥底面正多边形中心角半角值

G68 X0 Y0 R[#5+#9]; 坐标系旋转, 保证顶面一边垂直于 X 轴并对称于 X 轴

#13=#3; 给背吃刀量#13 赋初始值#3

WHILE[#13GE0] DO2; 循环 2 设定条件: 背吃刀量#13 不小于 0

#10=#2/2+#13/TAN[#12]; 计算任意切削层外接圆半径值

#11=#10*COS[#9]; 计算任意切削层棱边中心点 X 轴坐标值

#26=#13-#8*COS[#12]; 计算球心 Z 轴坐标

#15=#10+#8*SIN[#12]; 计算任意切削层刀具球心外接圆半径值

#14=#15*COS[#9]; 计算刀具切削到棱边中点时的 X 轴坐标

G00 X[#14+#18] Y0; 快速定位到切削准备点

Z[3-#26]; 快速下降到离背吃刀量 3mm 处

Y[#18+2]; 快速定位到螺旋切入起始点

G03 X#14 Y2 Z-#26 R#18 F#7; 螺旋切入

#20=-#9; 给夹角变量#20 赋初始值-#9


```

WHILE[#20GT-360] DO1; 循环 1 设定条件: 夹角变量#20 大于-360°
#24=#15*COS[#20]; 计算任意切削层刀具球心的 X 轴坐标
#25=#15*SIN[#20]; 计算任意切削层刀具球心的 Y 轴坐标
G01 X#24 Y#25; 直线切削到棱边顶点
#20=#20-2*#9; 夹角#20 递减一个棱锥底面正多边形中心角
END1; 循环 1 结束
G01 Y-2; 完成一层轮廓切削
G03 X[#14+#18] Y-[#18+2] Z[3-#26] R#18; 螺旋切出
G00 Y0; 返回加工准备点
#13=#13-#6; 背吃刀量变量#13 递减一个深度增量#6
END2; 循环 2 结束
G69; 坐标系旋转取消
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
%
```

4) 主程序如下:

```

%
O143;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6143 A75 B60 C8 I5 J15 K0.4 D1000 E8 R15; 调用宏程序 O6143,
                                                赋相关变量初始值
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

3. 带圆角外棱锥台加工

1) 分析: 如图 6-21 所示, 用立铣刀对底面外接圆直径为#1, 顶面外接圆直径为#2, 高#3, 棱边数#4, 底面圆角半径为#17 的外棱锥台进行粗加工。建立如图 6-21 所示坐标系, 以顶面中心为坐标系原点, 向上为 Z 轴正向。

首先利用坐标系旋转指令将顶面一边对称垂直于 X 轴正向, 见图 6-21 中 X'OY'坐标系。以#6 为深度变量, 从顶面每次下降一个#6 深度变量, 计算此时该层棱锥截面正多边形相关尺寸, 螺旋切入、半径补偿、顺时针走刀切削此多边形、

螺旋切出；一层一层如此反复，直至切削到底面。

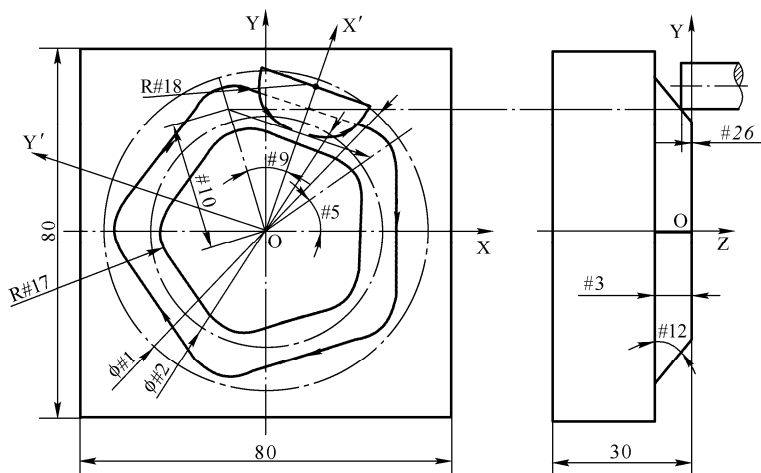


图 6-21 带圆角外棱锥台加工

2) 变量及说明:

- #1=[A]; 大端外接圆直径
- #2=[B]; 小端外接圆直径
- #3=[C]; 棱锥高
- #4=[I]; 棱锥棱边数
- #5=[J]; 第一顶点与 X 轴正向夹角
- #6=[K]; 每次背吃刀量增量
- #7=[D]; 进给量
- #8=[E]; 球头铣刀半径
- #17=[Q]; 顶面圆角半径
- #18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

① 用 G10 指令编写的程序。

%

O6144; 子程序号

#12=ATAN[2*#3]/[#1-#2]; 计算锥台棱线锥角

#9=180/#4; 计算棱锥底面正多边形中心角半角值

G68 X0 Y0 R[#5+#9]; 坐标系旋转, 保证顶面一边垂直于 X 轴并对称于 X 轴

#26=#6; 给背吃刀量#26 赋初始值一个切削增量#6

```

WHILE[#26LE#3] DO2; 循环 2 设定条件: 背吃刀量#6 不大于深度#3
#11=[#2/2]*COS[#9]; 计算任意切削层棱边中心点 X 轴坐标值
#20=#8/2+#26*COS[#9]/TAN[#12]; 计算半径补偿值修正值#20
G10 L12 P1 R#20; 将半径补偿值#20 赋给 D1 寄存器
G00 X[#11+#18] Y0; 快速定位到切削准备点
Z[3-#26]; 快速下降到离背吃刀量 3mm 处
G41 D1 Y[#18+2]; 建立半径补偿
G03 X#11 Y2 Z-#26 R#18 F#7; 螺旋切入
#15=-#9; 给棱边任意顶点与 X 轴夹角赋初始值-#9
WHILE[#15GT-360] DO1; 循环 1 设定条件: 底面顶点与 X 轴正向夹角大
于-360°
#24=#10*COS[#15]; 计算任意切削层棱边顶点的 X 轴坐标
#25=#10*SIN[#15]; 计算任意切削层棱边顶点的 Y 轴坐标
G01 X#24 Y#25 R#17; 直线切削到棱边顶点, 并倒圆
#15=#16-2*#9; 夹角#15 递减一个棱锥底面正多边形中心角
END1; 循环 1 结束
G01 Y-2; 完成一层轮廓切削
G03 X[#11+#18] Y-[#18+2] Z[3-#26] R#18; 螺旋切出, 抬刀 3mm
G00 G40 Y0; 取消半径补偿
#26=#26+#6; 背吃刀量累加一个深度增量#6
END2; 循环 2 结束
G69; 坐标系旋转取消
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
%
② 用固定半径补偿编写的程序。
%
O6145; 子程序号
#12=ATAN[2*#3]/[#1-#2]; 计算锥台棱线锥角
#9=180/#4; 计算棱锥底面正多边形中心角半角值
G68 X0 Y0 R[#5+#9]; 坐标系旋转, 保证顶面一边垂直于 X 轴并对称于
X 轴
#26=#6; 给背吃刀量#26 赋初始值一个切削增量#6
WHILE[#26LE#3] DO2; 循环 2 设定条件: 背吃刀量#6 不大于深度#3
#10=#2/2+#26/TAN[#12]; 计算任意层切削层正多边形外接圆半径

```

```

#11=#10*COS[#9]; 计算任意切削层棱边中心点 X 轴坐标值
#20=#17+#26/TAN[#12]; 计算圆角半径修正后值#20
G00 X[#11+#18] Y0; 快速定位到切削准备点
Z[3-#26]; 快速下降到离背吃刀量 3mm 处
G41 D1 Y[#18+2]; 建立半径补偿
G03 X#11 Y2 Z-#26 R#18 F#7; 螺旋切入
#15=-#9; 给棱边任意顶点与 X 轴夹角赋初始值-#9
WHILE[#15GT-360] DO1; 循环 1 设定条件: 底面顶点与 X 轴正向夹角大于-360°
#24=#10*COS[#15]; 计算任意切削层棱边顶点的 X 轴坐标
#25=#10*SIN[#15]; 计算任意切削层棱边顶点的 Y 轴坐标
G01 X#24 Y#25 R#20; 直线切削到棱边顶点, 并倒圆
#15=#16-2*#9; 夹角#15 递减一个棱锥底面正多边形中心角
END1; 循环 1 结束
G01 Y-2; 完成一层轮廓切削
G03 X[#11+#18] Y-[#18+2] Z[3-#26] R#18; 螺旋切出, 抬刀 3mm
G00 G40 Y0; 取消半径补偿
#26=#26+#6; 背吃刀量累加一个深度增量#6
END2; 循环 2 结束
G69; 坐标系旋转取消
G00 Z50; 快速上升至 Z=50
M99; 子程序结束
%
4) 主程序如下:
%
O144;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S400;
G65 P6144 A75 B60 C8 I5 J15 K1 D300 E16 Q8 R15; 调用宏程序 O6144, 赋
相关变量初始值
G00 G49 Z100;
X0 Y150;
M05;
M30;

```

%

4. 带圆角内棱锥台加工

1) 分析: 如图 6-22 所示, 用立铣刀对底面外接圆直径为#1, 顶面外接圆直径为#2, 高#3, 棱边数#4, 顶面圆角半径为#17 的内棱锥进行粗加工。建立如图 6-22 所示坐标系, 以顶面中心为坐标系原点, 向上为 Z 轴正向。

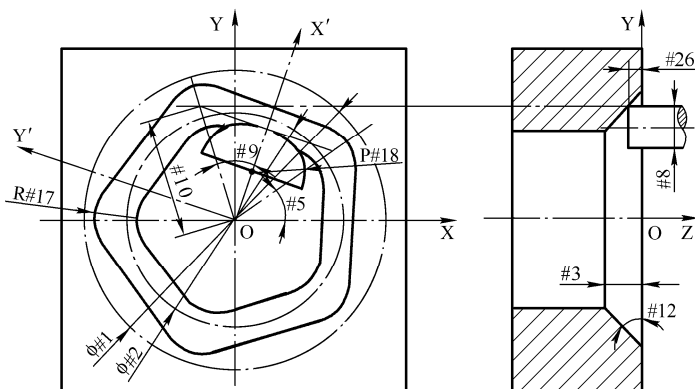


图 6-22 带圆角内棱锥台加工

首先利用坐标系旋转指令将顶面一边对称垂直于 X 轴正向, 见图 6-22 中 X'OY'坐标系。以#6 为深度变量, 从顶面每次下降一个#6 深度变量, 计算此时该层棱锥截面正多边形相关尺寸, 螺旋切入、半径补偿、逆时针走刀切削此多边形、螺旋切出; 一层一层如此反复, 直至切削到底面。

2) 变量及说明:

#1=[A]; 大端外接圆直径

#2=[B]; 小端外接圆直径

#3=[C]; 棱锥高

#4=[I]; 棱锥棱边数

#5=[J]; 第一顶点与 X 轴正向夹角

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

#8=[E]; 刀具直径

#17=[Q]; 顶面圆角半径

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

① 用 G10 指令编写的程序。

%

O6146; 子程序号

#12=ATAN[2*#3]/[#1-#2]; 计算锥台棱线锥角

#9=180/#4; 计算棱锥底面正多边形中心角半角值

G68 X0 Y0 R[#5+#9]; 坐标系旋转, 保证顶面一边垂直于 X 轴并对称于 X 轴

#26=#6; 给背吃刀量#26 赋初始值一个切削增量#6

WHILE[#26LE#3] DO2; 循环 2 设定条件: 背吃刀量#6 不大于深度#3

#11=[#2/2]*COS[#9]; 计算任意切削层棱边中心点 X 轴坐标值

#20=#8/2+#26*COS[#9]/TAN[#12]; 计算半径补偿值修正值#20

G10 L12 P1 R#20; 将半径补偿值#20 赋给 D1 寄存器

G00 X[#11-#18] Y0; 快速定位到切削准备点

Z[3-#26]; 快速下降到离背吃刀量 3mm 处

G41 D1 Y-[#18+2]; 建立半径补偿

G03 X#11 Y-2 Z-#26 R#18 F#7; 螺旋切入

#15=#9; 给棱边任意顶点与 X 轴夹角赋初始值#9

WHILE[#15LT360] DO1; 循环 1 设定条件: 夹角变量#15 小于 360°

#24=#10*COS[#15]; 计算任意切削层棱边顶点的 X 轴坐标

#25=#10*SIN[#15]; 计算任意切削层棱边顶点的 Y 轴坐标

G01 X#24 Y#25 R#17; 直线切削到棱边顶点, 并倒圆

#15=#15+2*#9; 夹角#15 递增一个棱锥底面正多边形中心角

END1; 循环 1 结束

G01 Y2; 完成一层轮廓切削

G03 X[#11-#18] Y[#18+2] Z[3-#26] R#18; 螺旋切出, 抬刀 3mm

G00 G40 Y0; 取消半径补偿

#26=#26+#6; 背吃刀量累加一个深度增量#6

END2; 循环 2 结束

G69; 坐标系旋转取消

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

② 用固定半径补偿编写的程序。

%

O6147; 子程序号

#12=ATAN[2*#3]/[#1-#2]; 计算锥台棱线锥角

#9=180/#4; 计算棱锥底面正多边形中心角半角值

G68 X0 Y0 R[#5+#9]; 坐标系旋转, 保证顶面一边垂直于 X 轴并对称于 X 轴

#26=#6; 给背吃刀量#26 赋初始值一个切削增量#6

WHILE[#26LE#3] DO2; 循环 2 设定条件: 背吃刀量#6 不大于深度#3

#10=#2/2-#26/TAN[#12]; 计算任意层切削层正多边形外接圆半径

#11=#10*COS[#9]; 计算任意切削层棱边中心点 X 轴坐标值

#20=#8/2-#26*COS[#9]/TAN[#12]; 计算圆角半径修正后值#20

G00 X[#11-#18] Y0; 快速定位到切削准备点

Z[3-#26]; 快速下降到离背吃刀量 3mm 处

G41 D1 Y-[#18+2]; 建立半径补偿

G03 X#11 Y-2 Z-#26 R#18 F#7; 螺旋切入

#15=#9; 给棱边任意顶点与 X 轴夹角赋初始值#9

WHILE[#15LT360] DO1; 循环 1 设定条件: 夹角变量#15 小于 360°

#24=#10*COS[#15]; 计算任意切削层棱边顶点的 X 轴坐标

#25=#10*SIN[#15]; 计算任意切削层棱边顶点的 Y 轴坐标

G01 X#24 Y#25 R#20; 直线切削到棱边顶点, 并倒圆

#15=#15+2*#9; 夹角#15 递增一个棱锥底面正多边形中心角

END1; 循环 1 结束

G01 Y2; 完成一层轮廓切削

G03 X[#11-#18] Y[#18+2] Z[3-#26] R#18; 螺旋切出, 抬刀 3mm

G00 G40 Y0; 取消半径补偿

#26=#26+#6; 背吃刀量累加一个深度增量#6

END2; 循环 2 结束

G69; 坐标系旋转取消

G00 Z50; 快速上升至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O146;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S400;

G65 P6146 A75 B60 C8 I5 J35 K1 D300 E16 Q15 R15; 调用宏程序 O6146, 赋相关变量初始值

```
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.1.5 实例五球体加工

1. 凸球面粗加工（立铣刀 $\Phi 16$ 、由上而下）

1) 分析：如图 6-23 所示，球冠半径 $\#1$ ，高 $\#3$ ，现在用直径 $\#2$ 立铣刀进行粗加工，毛坯为圆柱体。建立如图 6-23 所示坐标系。

首先计算球冠底面半径： $\#11 = \text{SQRT}[\#1 * \#1 - [\#1 - \#3] * [\#1 - \#3]]$ ，每层切削时的行间距： $\#8 = 0.8 * \#2$ ，以背吃刀量 $\#26$ 为自变量， $\#6$ 为深度递增量， $\#26$ 初始值为 $\#6$ ，以后每次增加 $\#6$ 直到 $\#3$ ，一层一层由上而下逐一进行余量切削，加工路线轨迹： $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow B' \rightarrow C' \rightarrow D' \rightarrow E'$ 。

具体加工编程思路：首先快速定位到 $([\#11 + 2 + 0.5 * \#2], 0, 5)$ 准备加工位置，下降到第一次切削层，计算该层球冠底面半径 $\#12 = \text{SQRT}[\#1 * \#1 - [\#1 - \#26] * [\#1 - \#26]]$ ，得到该层切削余量为 $\#11 - \#12$ ，如果该余量小于刀具直径，则一次完成切削，刀具中心位置： $\#20 = \#12 + 0.5 * \#2$ ；如果它大

于刀具直径，则用它除以 $\#8$ 算出需要水平切削次数，从外向内逐一切削，最外层刀具中心 X 轴坐标 $\#11 - 0.3 * \#2$ ，顺时针整圆铣削，再向内直线进给一个行间距 $\#8$ ，接着顺时针整圆铣削，刀具中心最小尺寸不小于 $\#20 = \#12 + 0.5 * \#2$ ，直到完成该层余量切削。完成这一层切削后，抬刀 2mm，快速返回准备加工位置，再下降一个背吃刀量递增量 $\#6$ ，重新计算相关值，完成此层的加工，一直到最后整个球冠的粗加工。

2) 参数及说明：

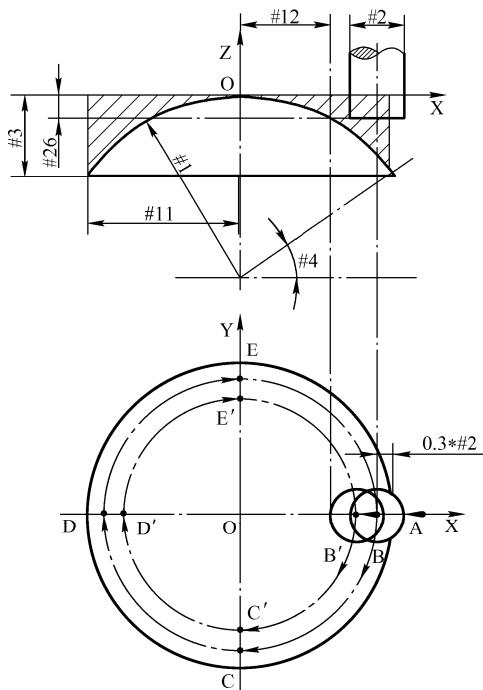


图 6-23 凸球面粗加工

#1=[A]; 球半径

#2=[B]; 刀具直径

#3=[C]; 球冠高度

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

3) 宏程序如下:

%

O6151; 子程序号

#8=0.8*#2; 行间距

#11=SQRT[#1*#1-[#1-#3]*[#1-#3]]; 计算球冠底面半径

G00 X[#11+2+0.5*#2] Y0; 快速定位到球冠右侧距中心轮廓 2mm 处
Z5; 快速下降到 Z=5

#9=FIX[#3/#6]; 计算深度切削次数

#13=0; 给背吃刀量计数器赋初始值 0

WHILE[#13LE#9] DO2; 循环 2 设定条件: 计数器值不大于#9

#26=#6*[#13+1]; 计算每次背吃刀量的深度值

#12=SQRT[#1*#1-[#1-#26]*[#1-#26]]; 计算任意深度对应的平面圆半径

#20=#12+0.5*#2; 刀具中心最小极限尺寸坐标计算

#10=FIX[[(#11-#12)/#8]]; 计算切削余量及水平切削次数

G00 Z-#26; 快速下降到背吃刀量位置

#14=0; 水平切削次数计数器值赋初始值为 0

WHILE[#14LE#10] DO1; 循环 1 设定条件: 计数器值不大于计算的水平切
削次数

#24=#11-0.3*#2-#8*#14; 计算每轮刀具中心坐标值

IF[#24LT#20] THEN#24=#20; 限定刀具中心坐标值不小于其最小极限尺寸

IF[#11-#12LT#2] GOTO10; 如果水平切削余量小于刀具直径, 跳转到顺序
号 10 程序段

GOTO20; 跳转到顺序号 20 程序段

N10 #24=#20; 顺序号 10 程序段: 将#20 值赋给#24

N20 G01 X#24 F#7; 顺序号 20 程序段: 直线切削至相应轮次 X 轴坐标
值处

G02 I-#24; 顺时针整圆圆弧切削

#14=#14+1; 水平切削次数计数器值累加 1

END1; 循环 1 结束

G00 Z[2-#26]; 抬刀 2mm

X[#11+2+0.5*#2] Y0; 返回刚才准备点

#13=#13+1; 背吃刀量计数器累加 1

END2; 循环 2 结束

G00 Z50; 快速上升到 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O151;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S400;

G65 P6151 A35 B16 C15 K1.5 D300; 调用宏程序 O6151, 给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 凸球面精加工 (球头铣刀、由上而下)

1) 分析: 如图 6-24 所示, 用球头铣刀加工此球冠, 先计算出该球冠底面圆轮廓点与球心连线跟水平面之间的夹角#4 和球头铣刀球心在加工时的轨迹半径#20。从#4 到 90° , 每次增加#6, 逐一计算刀具加工在此处球心 X、Z 坐标值 (在 X、Z 坐标平面上)。

刚开始刀具快速定位在准备位置, 螺旋切入至深度#3, 不必半径补偿, 以刀具球心作为参考点计算坐标, 顺时针整圆铣削球冠截面圆轮廓, 再螺旋切出, 夹角递增一个#6 后, 重新计算球心 X、Z 坐标, 接着重复进行螺旋切入、整圆铣削、螺旋切出, 一层一层由下而上直到完成整个球

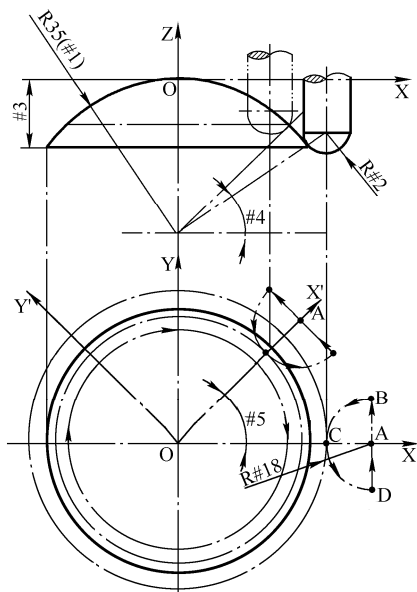


图 6-24 凸球面精加工

冠加工。

为了避免切入切出点过于集中而给工件表面留下明显痕迹，在水平面上多设了一个角度变量#5，每当铣削一水平层后， $\#4+\#6$ 则 $\#5+2*\#6$ ，切入切出点在水平方向上逆时针转过 $2*\#6$ ，并利用坐标系旋转指令 G68 简化坐标计算。

2) 参数及说明：

#1=[A]；球半径

#2=[B]；球头铣刀半径

#3=[C]；球冠高度

#6=[K]；每次背吃刀量增量

#7=[D]；进给量

#18=[R]；1/4 圆弧切入切出半径

3) 宏程序如下：

%

O6152；子程序号

#20=#1+#2；计算中心距

#5=0；给水平旋转角度赋初始值为 0

G00 X#20 Y0；快速定位到起始位置

Z5；快速下降到 Z=5

#13=SQRT[#1*#1-[#1-#3]*[#1-#3]]；计算球冠底面圆半径

#4=ATAN[#1-#3]/[#13]；计算球冠底面圆周与 X 轴交点对应的夹角即起始角度

WHILE[#4LE90] DO1；循环 1 设定条件：夹角#4 不大于 90°

#24=#20*COS[#4]；计算相应点刀具球心 X 坐标值

#26=#20*SIN[#4]-#1；计算相应点刀具球心 Z 坐标值

G68 X0 Y0 R#5；坐标系旋转角度#5

G00 X[#24+#18] Y0；快速定位到螺旋切入准备位置

Z[2+#26]；快速定位到距背吃刀量 2mm 处

Y#18；快速定位到螺旋切入起始点

G03 X#24 Y0 Z#26 R#18 F#7；螺旋切入

G02 I-#24；顺时针整圆圆弧切削

G03 X[#24+#18] Y-#18 Z[2+#26] R#18；螺旋切出

G00 Y0；返回准备位置

#4=#4+#6；夹角累加一个角度增量#6

#5=#5+2*#6；水平旋转角度累加 2 个角度增量#6

G69；坐标系旋转取消

END1; 循环 1 结束

G00 Z50; 快速上升到 Z=50

M99; 子程序

%

4) 主程序如下:

%

O152;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6152 A35 B8 C15 K5 D1000 R10; 调用宏程序 O6152, 赋相关变量初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

3. 凸球台粗加工 (立铣刀、由上而下)

1) 分析: 如图 6-25 所示, 它与图 6-23 的主要区别就是球冠下面多了一个尺寸大一些的圆柱平台, 其他完全相同, 所以加工过程可以完全参照图 6-23 进行相关计算与编程。

2) 参数及说明:

#1=[A]; 球半径

#2=[B]; 刀具直径

#3=[C]; 球冠直径

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

3) 宏程序如下:

%

O6153; 子程序号

#8=0.8*#2; 行间距

#5=#1-SQRT[#1*#1-#3*#3*0.25]; 计算球冠高

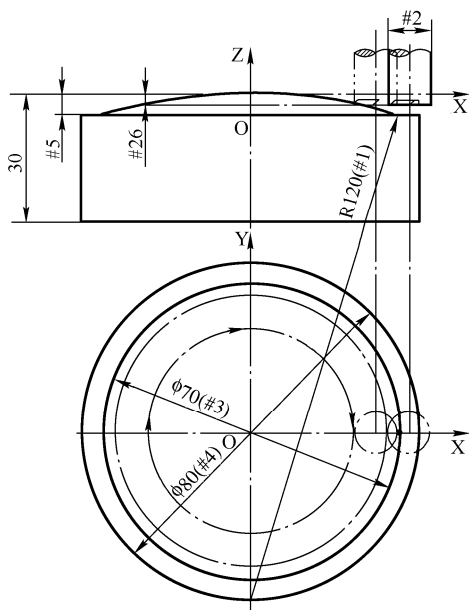


图 6-25 凸球台粗加工

```

G00 X[#4/2+#2/2+2] Y0; 刀具快速定位到毛坯右侧距其轮廓 2mm 处
Z5; 快速下降到 Z=5
#9=FIX[#5/#6]; 计算深度切削次数
#11=0; 给背吃刀量计数器赋初始值 0
WHILE[#11LE#9] DO2; 循环 2 设定条件: 计数器值不大于#9
#26=#6*[#11+1]; 计算每次背吃刀量的深度值
IF[#26GT#5] THEN#26=#5; 限定背吃刀量最大不超过#5
#12=SQRT[#1*#1-[#1-#26]*[#1-#26]]; 计算每层球冠底面半径
#20=#12+0.5*#2; 刀具中心最小极限尺寸坐标计算
#10=FIX[[(#4/2-#12)/#8]; 计算切削余量及水平切削次数
G00 Z-#26; 快速下降到背吃刀量位置
#14=0; 水平切削次数计数器值赋初始值为 0
WHILE[#14LE#10] DO1; 循环 1 设定条件: 计数器值不大于计算的水平切
                        削次数
#24=#4/2-0.3*#2-#8*#14; 计算每轮刀具中心坐标值
IF[#24LT#20] THEN#24=#20; 限定刀具中心坐标值不小于其最小极限尺寸
IF[#4/2-#12LT#2] GOTO10; 如果水平切削余量小于刀具直径, 跳转到顺序
                        号 10 程序段
GOTO20; 跳转到顺序号 20 程序段
N10 #24=#20; 顺序号 10 程序段: 将#20 值赋给#24
N20 G01 X#24 F#7; 顺序号 20 程序段: 直线切削至相应轮次 X 轴坐标
                        值处
G02 I-#24; 顺时针整圆圆弧切削
#14=#14+1; 水平切削次数计数器值累加 1
END1; 循环 1 结束
G00 Z[2-#26]; 抬刀 2mm
X[#4/2+#2/2+2] Y0; 返回刚才准备点
#11=#11+1; 背吃刀量计数器累加 1
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50
M99; 子程序结束
%
4) 主程序如下:
%
O153;

```

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S400;

G65 P6153 A35 B16 C70 K5 D300; 调用宏程序 O6153, 赋相关变量
初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

4. 凸球台精加工（立铣刀、由下而上）

1) 分析: 如图 6-26 所示, 它与图 6-24 的主要区别就是球冠下面多了一个尺寸大一些的圆柱平台, 其他完全相同, 但是它不能用球头铣刀而只能用立铣刀, 加工过程可以参照图 6-24 的加工思路, 具体坐标计算有所区别, 最好用半径补偿, 简化坐标计算。

2) 参数及说明:

#1=[A]; 球半径

#2=[B]; 刀具直径

#3=[C]; 球冠底面直径

#6=[K]; 角度增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6154; 子程序号

#4=#1-SQRT[#1*#1-#3*#3/4]; 计算球冠高

#5=ATAN[#1-#4]/[#3/2]; 计算球冠中心角

G00 X[#3+#18] Y0; 快速定位到加工准备位置

#8=0; 给水平旋转角度赋初始值 0

WHILE[#5LT90] DO1; 循环 1 设定条件: 夹角变量#5 小于 90°

G68 X0 Y0 R#8; 坐标系旋转#8

#24=#1*COS[#5]; 计算球面轮廓点 X 轴坐标

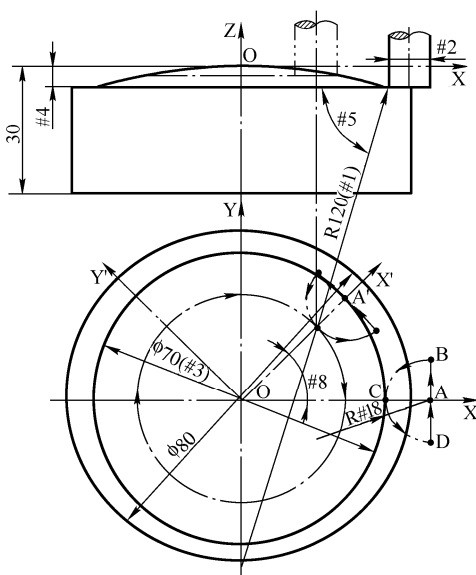


图 6-26 凸球台精加工

```

#26=#1*[1-SIN[#5]]; 计算球面轮廓点 Z 轴坐标
G00 X[#24+#18] Y0; 快速定位到半径补偿开始点
Z[2-#26]; 快速下降到距当前层 2mm 处
G41 D1 X[#24+#18] Y#18; 半径补偿
G03 X#24 Y0 Z-#26 R#18 F#7; 螺旋切入
G02 I-#24; 顺时针整圆切削
G03 X[#24+#18] Y-#18 Z[2-#26] R#18; 螺旋切出
G00 G40 Y0; 取消半径补偿
#8=#8+2*#6; 水平旋转角度#8 累加 2 倍的角度递增量#6
#5=#5+#6; 夹角#5 累加一个角度递增量#6
G69; 坐标系旋转取消
END1; 循环 1 结束
G00 Z50; 快速上升到 Z=50
M99; 子程序
%
```

4) 主程序如下:

%

O154;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6154 A120 B16 C70 K5

D1000; 调用宏程序 O6154, 赋相关变量初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

5. 凹球面粗加工 (立铣刀、由上而下)

1) 分析: 如图 6-27 所示, 建立坐标系, 先计算出立铣刀所能切削的最大深度#8, 加工思路是: 从上而下, 先从上下降一个深度递增量#6, 水平径向从中心向球面轮廓, 根据该层余量大小, 行间距为#10=0.8*#2, 分配

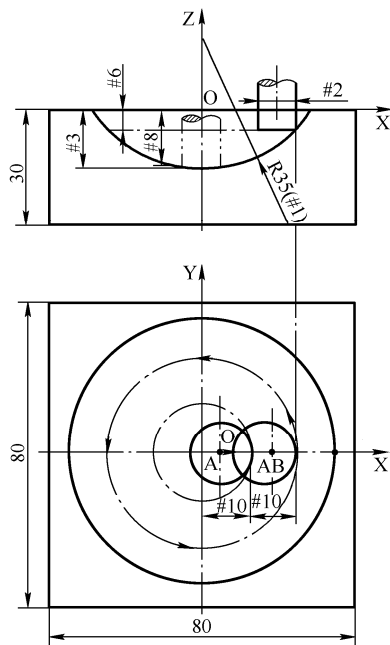


图 6-27 凹球面粗加工

切削次数，每次先由内而外直线切削计算好的径向半径处（ $\#24=0.3*\#2+\#10*\#9$ ），然后水平逆时针整圆铣削成圆柱槽孔，深度方向则由 $\#6$ 逐层切削到深度 $\#8$ 。

2) 参数及说明：

$\#1=[A]$ ；球半径

$\#2=[B]$ ；刀具直径

$\#3=[C]$ ；球冠高度

$\#6=[K]$ ；深度递增量

$\#7=[D]$ ；进给量

$\#18=[R]$ ；1/4圆弧切入切出半径

3) 宏程序如下：

%

O6155；子程序号

$\#10=0.8*\#2$ ；计算行间距

G00 X0 Y0；快速定位到（0，0）

Z5；快速下降到Z5处

$\#23=\#1-\#3$ ；中间值寄存

$\#8=\text{SQRT}[\#1*\#1-\#2*\#2/4]-\#23$ ；计算立铣刀最大加工深度

$\#26=\#6$ ；给背吃刀量变量 $\#26$ 赋初始值一个深度递增量 $\#6$

WHILE $[\#26\text{LE}\#8]$ DO2；循环2设定条件：背吃刀量变量 $\#26$ 不大于最大加工深度 $\#8$

$\#30=[\#23+\#26]*[\#23+\#26]$ ；中间值寄存

$\#21=\text{SQRT}[\#1*\#1-\#30]$ ；计算任意深度层水平加工余量半径

$\#5=\text{FIX}[\#21/\#10]$ ；计算水平切削次数

$\#22=\#21-\#2/2$ ；设定刀具在任意深度最大X轴方向坐标

G01 Z $-\#26$ F $[0.3*\#7]$ ；直线切削到背吃刀量 $\#26$

$\#9=0$ ；给水平切削次数计数器变量 $\#9$ 赋初始值0

WHILE $[\#9\text{LE}\#5]$ DO1；循环1设定条件：水平切削次数计数器变量 $\#9$ 不大于所需水平切削次数 $\#5$

$\#24=0.3*\#2+\#10*\#9$ ；计算水平切削任意轮次刀具中心X轴坐标

IF $[\#24\text{GT}\#22]$ THEN $\#24=\#22$ ；如果刀具中心X轴坐标大于其最大坐标，则限定它最大等于 $\#22$

G01 X $\#24$ F $\#7$ ；水平直线切削到X $\#24$

G03 I $-\#24$ ；逆时针整圆切削

$\#9=\#9+1$ ；水平切削次数计数器变量 $\#9$ 累加1

END1；循环1结束

G00 Z[2-#26]; 抬刀 2mm

X0 Y0; 快速定位到 (0, 0)

#26=#26+#6; 背吃刀量#26 累加一个深度递增量#6

END2; 循环 2 结束

G00 Z50; 快速上升到 Z=50

M99; 子程序

%

4) 主程序如下:

%

O155;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6155 A35 B16 C15 K1.5 D300; 调用宏程序 O6155, 赋相关变量初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

6. 凹球面精加工 (球头铣刀、由下而上)

1) 分析: 如图 6-28 所示, 凹球面精加工, 此处不能用立铣刀进行加工, 因为它不能加工球面底面轮廓, 所以这里用半径小于球面半径的球头铣刀进行加工。

2) 参数及说明:

#1=[A]; 球半径

#2=[B]; 刀具直径

#3=[C]; 球冠高度

#6=[K]; 角度递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

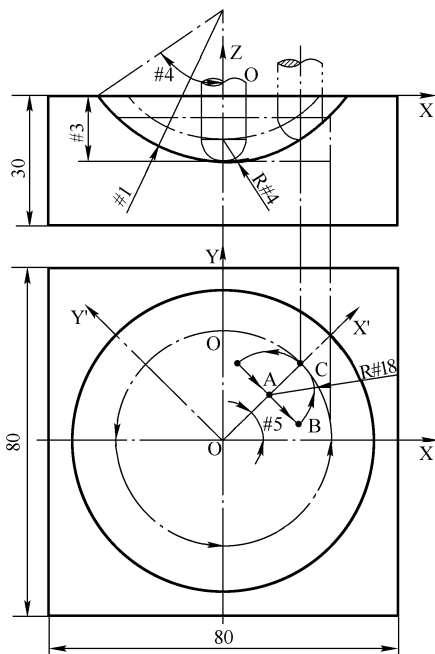


图 6-28 凹球面精加工

%

O6156; 子程序号

#20=#1-#3; 计算凹球面球心到其最大截面圆距离

#21=SQRT[#1*#1-#20*#20]; 计算凹球面最大截面圆半径

#4=ATAN[#21]/[#20]; 计算夹角变量#8 的最大极限角度#4

G00 X0 Y0; 快速定位到球心位置

Z[2-#3+#2]; 快速下降到离球心正下方球面 2mm 处

G01 Z[#2-#3] F[0.3*#7]; 直线切削到凹球面中心最低点

#8=0; 给夹角变量#8 赋初始值 0

#5=0; 给水平夹角变量#5 赋初始值 0

WHILE[#8LE#4] DO1; 循环 1 设定条件: 夹角变量#8 不大于#4

G68 X0 Y0 R#5; 坐标系统绕 (0, 0) 旋转角度#5

#24=[#1-#2]*SIN[#8]; 计算相对于夹角变量#8 刀具球心的 X 轴坐标

#26=[#1-#2]*COS[#8]-#20; 计算相对于夹角变量#8 刀具球心的 Z 轴坐标

G01 X#24 Z-#26 Y0 F#7; 螺旋切削到整圆切削开始点

G03 I-#24; 逆时针整圆切削

#8=#8+#6; 夹角变量#8 累加一个角度递增量#6

#5=#5+2*#6; 水平角度变量#5 累加两个角度递增量#6

G69; 坐标系旋转取消

END1; 循环 1 结束

G00 Z50; 快速上升到 Z=50

M99; 子程序

%

4) 主程序如下:

%

O156;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6156 A35 B8 C15 K5 D1000; 调用宏程序 O6156, 赋相关变量
初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

6.1.6 实例六椭圆球面加工

1. 外椭圆球面粗加工

1) 分析: 如图 6-29 所示, 用直径#2 的立铣刀进行外椭圆球面粗加工。

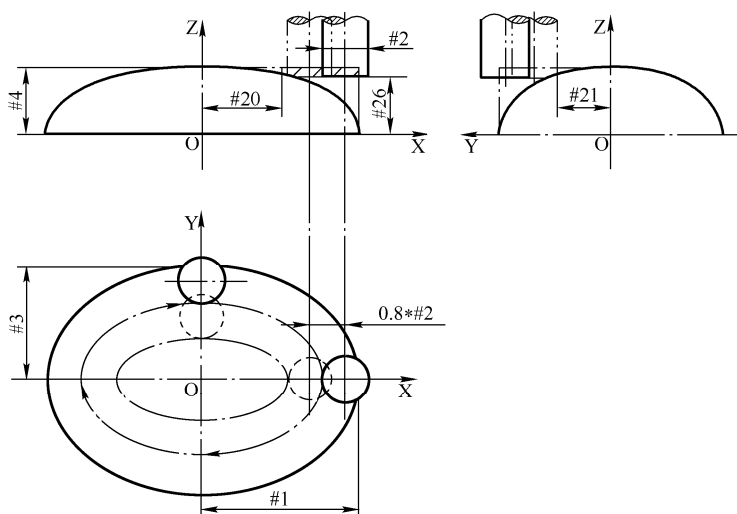


图 6-29 外椭圆球面粗加工

由上而下, 从背吃刀量 $\#26=\#4-\#6$ 开始, 再到 $\#26=0$, 行间距 $\#8=0.8*\#2$, 每层先计算该层切削余量, X、Y 轴向余量不同时, 取两者大值去除以行间距得到此层需要切削的次数, 由外向内环切椭圆平面; 抬刀返回加工起始点, 下刀至下一加工层, 继续计算需要切削的次数, 继而再完成此层椭圆平面的加工, 这样一层一层加工。

2) 变量及说明:

$\#1=[A]$; 椭圆球 X 轴方向半轴长

$\#2=[B]$; 立铣刀直径

$\#3=[C]$; 椭圆球 Y 轴方向半轴长

$\#4=[I]$; 椭圆球 Z 轴方向半轴长

$\#6=[K]$; 深度增量

$\#7=[D]$; 进给量

$\#17=[Q]$; 角度递增量

3) 宏程序如下:

%

O6161; 子程序号

G00 X0 Y[#3+#2/2+2]; 快速定位到加工准备位置

Z[#4+2]; 快速下降到离工件上平面 2mm 处

#8=0.8*#2; 计算行间距

#26=#4-#6; 给背吃刀量变量#26 赋初始值, 第一刀背吃刀量#6

WHILE[#26GE0] DO3; 循环 3 设定条件: 背吃刀量变量#26 不小于 0

G01 Z#26 F#7; 直线切削到第一刀深度

#30=SQRT[1-#26*#26/[#4*#4]]; 中间计算值寄存在变量#30 中

#20=#30*#1; 计算在深度#26 时, 截面椭圆 X 轴向半轴长

#21=#30*#3; 计算在深度#26 时, 截面椭圆 Y 轴向半轴长

IF[#1-#20GE#3-#21] THEN#9=#1-#20; 如果 X 轴向余量大于等于 Z 轴向余量, 则取 X 轴向余量

IF[#1-#20LT#3-#21] THEN#9=#3-#21; 如果 X 轴向余量小于 Z 轴向余量, 则取 Z 轴向余量

#10=FIX[#9/#8]; 计算水平切削次数

#15=0; 给水平切削次数计数器赋初始值 0

WHILE[#15LE#10] DO2; 循环 2 设定条件: 计数器值#15 不大于#10

#11=#1-0.3*#2-#8*#15; 计算每次水平切削刀具中心 X 轴向半轴长

#31=#20+#2/2; 计算刀具中心 X 轴半轴长最小极限尺寸

IF[#11LT#31] THEN#11=#31; 限定刀具中心 X 轴向半轴长不小于最小极限尺寸

#32=#21+#2/2; 计算每次水平切削刀具中心 Y 轴向半轴长

#12=#2-0.3*#2-#8*#15; 限定刀具中心 Y 轴半轴长最小极限尺寸

IF[#12LT#32] THEN#12=#32; 限定刀具中心 Y 轴向半轴长不小于最小极限尺寸

#16=90; 设定水平切削椭圆角度变量#16 初始值为 90°

WHILE[#16GE-270] DO1; 循环 1 设定条件: 角度变量#16 不小于-270°

#24=#11*COS[#16]; 计算刀具中心 X 轴向坐标

#25=#12*SIN[#16]; 计算刀具中心 Y 轴向坐标

G01 X#24 Y#25 F#7; 直线切削到刚计算好的切削点 (#24, #25)

#16=#16-#17; 角度变量#16 累减一个角度递增量#17

END1; 循环 1 结束

#15=#15+1; 水平切削轮次计数器值#15 累加 1

END2; 循环 2 结束

G00 Z[2+#26]; 抬刀 2mm

X0 Y[#3+#2/2+2]; 快速定位到切削准备位置

#26=#26-#6; 背吃刀量递减一个深度递增量#6

END3; 循环 3 结束

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

O0161;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S400;

G65 P6161 A35 B16 C25 I15 K1.5 D300 Q10; 调用宏程序 O6161,
赋相关变量初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 外椭圆球面精加工

(1) 立铣刀

1) 分析: 如图 6-30 所示, 用直径 #2 的立铣刀进行外椭圆球面的精加工。

加工思路: 由上而下, 由四周向中心扩展完成整个球面的精加工。

加工起始高度为 0, 以 #6 作为深度递增量逐层上升, 逐层加工先建立半径补偿, 圆弧切入切出, 再顺时针加工此椭圆平面, 一层一层加工, 直至完成所有加工。

2) 变量及说明:

#1=[A]; 椭圆球 X 轴方向半轴长

#2=[B]; 立铣刀直径

#3=[C]; 椭圆球 Y 轴方向半轴长

#4=[I]; 椭圆球 Z 轴方向半轴长

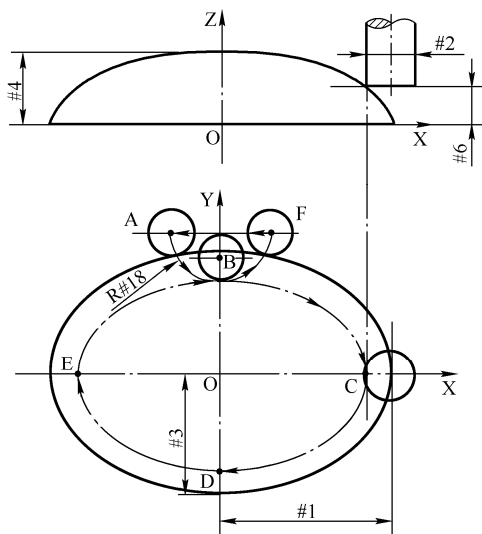


图 6-30 外椭圆球面精加工 (1)

#6=[K]; 深度增量

#7=[D]; 进给量

#17=[Q]; 角度递增量

#18=[D]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6162; 子程序号

G00 X0 Y[#3+#18]; 快速定位到加工准备位置

Z0; 快速下降到加工起始深度

#26=0; 给背吃刀量变量#26 赋初始值 0

WHILE[#26LE#4] DO2; 循环 2 设定条件: 背吃刀量变量#26 不大于#4

G01 Z#26 F#7; 直线切削到第一刀深度

#30=SQRT[1-#26*#26/[#4*#4]]; 中间计算值寄存在变量#30 中

#20=#30*#1; 计算在深度#26 时, 截面椭圆 X 轴向半轴长

#21=#30*#3; 计算在深度#26 时, 截面椭圆 Y 轴向半轴长

G41 D1 X-#18 Y[#21+#18]; 半径补偿

G03 X0 Y#21 R#18; 圆弧切入

#16=90; 给角度变量#16 赋初始值 90°

WHILE[#16GE-270] DO1; 循环 1 设定条件: 角度变量#16 不小于-270°

#24=#20*COS[#16]; 计算刀具中心 X 轴坐标

#25=#21*SIN[#16]; 计算刀具中心 Y 轴坐标

G01 X#24 Y#25 F#7; 直线切削到计算的刀具中心点 (#24, #25)

#16=#16-#17; 角度变量累减#17

END1; 循环 1 结束

G03 X#18 Y[#21+#18] R#18; 圆弧切出

G00 Z[2+#26]; 抬刀 2mm

G40 X0; 取消半径补偿

#26=#26+#6; 背吃刀量累加#6

END2; 循环 2 结束

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

O0162;

```
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6162 A35 B16 C25 I15 K0.5 D1000 Q10 R10; 调用宏程序 O6162,
                                                赋相关变量初始值
```

```
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

(2) 球头铣刀

1) 分析: 如图 6-31 所示, 用球头铣刀进行外椭圆球面精加工。

加工思路: 由下而上, 由四周向中心扩展完成整个球面的精加工, X、Y、Z 轴向的半轴长+刀具半径, 此值为球头铣刀参考点球心加工时三向的半轴长。

加工起始高度为 0, 以#6 作为深度递增量逐层上升, 逐层加工前先计算好该层 X、Y 轴向半轴长, 再顺时针加工此椭圆平面, 一层一层加工, 直至完成所有加工。

2) 变量及说明:

#1=[A]; 椭圆球 X 轴方向半轴长

#2=[B]; 球头铣刀半径

#3=[C]; 椭圆球 Y 轴方向半轴长

#4=[I]; 椭圆球 Z 轴方向半轴长

#6=[K]; 深度增量

#7=[D]; 进给量

#17=[Q]; 角度递增量

#18=[D]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6163; 子程序号

#11=#1+#2; 确定球头铣刀球心 X 轴半轴长

#13=#3+#2; 确定球头铣刀球心 Y 轴半轴长

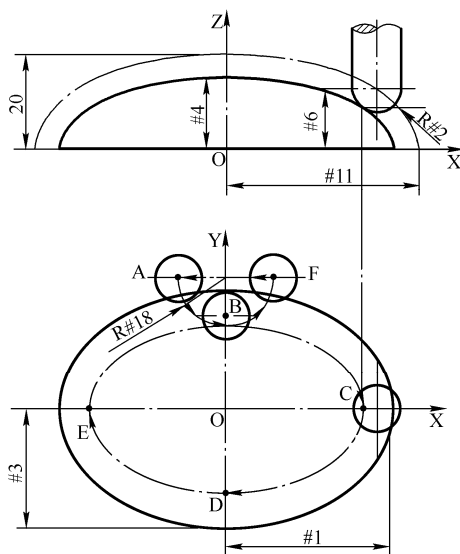


图 6-31 外椭圆球面精加工 (2)

```
#14=#4+#2; 确定球头铣刀球心 Z 轴半轴长
G00 X0 Y[#13+#18]; 快速定位到加工准备位置
Z0; 快速下降到切削底面
#26=0; 给深度变量#26 赋初始值 0
WHILE[#26LE#14] DO2; 循环 2 设定条件: 深度变量#26 不大于球心 Z 轴
    半轴长
G01 Z#26 F#7; 直线切削到深度#26
#30=SQRT[1-#26*#26/#14*#14]; 中间计算值寄存在变量#30 中
#20=#30*#11; 计算相应层球心 X 轴半轴长
#21=#30*#13; 计算相应层球心 Y 轴半轴长
X-#18 Y[#21+#18]; 直线切削到圆弧切入开始点
G03 X0 Y#21 R#18; 圆弧切入
#16=90; 给角度变量#16 赋初始值 90°
WHILE[#16GE-270] DO1; 循环 1 设定条件: 角度变量#16 不小于-270°
#24=#20*COS[#16]; 计算球心 X 轴坐标
#25=#21*SIN[#16]; 计算球心 Y 轴坐标
G01 X#24 Y#25 F#7; 直线切削到新切削点 (#24, #25)
#16=#16-#17; 角度变量#16 累减#17
END1; 循环 1 结束
G03 X#18 Y[#21+#18] R#18; 圆弧切出
G00 Z[2+#26]; 抬刀 2mm
X0; 返回准备点
#26=#26+#6; 背吃刀量累加一个深度递增量#6
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50 处
M99; 子程序结束
%
4) 主程序如下:
%
O0163;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6163 A35 B8 C25 I15 K0.5 D1000 Q10 R10; 调用宏程序 O6163,
    赋相关变量初始值
```


G00 G49 Z100;

X0 Y150;

M05;

M30;

%

3. 内椭圆球面加工

(1) 立铣刀

1) 分析: 如图 6-32 所示, 用立铣刀进行内椭圆球面的粗加工, 内椭圆球 X、Y、Z 轴向的半轴长分别为 #1、#3、#4; 立铣刀直径为 #2, 建立坐标系。

首先计算好刀具最多能加工的深度 $\#18 = \text{SQRT}[1 - (\#2 \times \#2 / 4) / (\#3 \times \#3)] \times \#4$, 行间距 $\#8 = 0.8 \times \#2$, 由上而下, 由中心向外, 逐层椭圆面铣削, 开始加工时刀具快速定位在 (0, 0, 2), 缓慢切削到

深度 -#6, 计算此层切削余量, 能几次完成切削加工, 从最里 ($0.3 \times \#2$, 0, -#6) 开始, 计算好 X、Y 轴向半轴长, 然后加工此椭圆; 再向外切削一个行间距 #8, 相应计算 X、Y 轴向半轴长, 接着再加工此椭圆, 依此类推, 直到加工到刚才算好的该层最大半轴长, 从而完成该层内椭圆球面的加工。

完成一层加工后, 抬刀快速返回到 X0 Y0, 接着再切削一个深度递增量 #6, 进行相关计算与加工, 这样逐层从上而下, 完成内椭圆球面的粗加工。

2) 变量及说明:

#1=[A]; 椭圆球 X 轴方向半轴长

#2=[B]; 立铣刀直径

#3=[C]; 椭圆球 Y 轴方向半轴长

#4=[I]; 椭圆球 Z 轴方向半轴长

#6=[K]; 深度增量

#7=[D]; 进给量

#17=[Q]; 角度递增量

3) 宏程序如下:

%

O6164; 子程序号

G00 X0 Y0; 快速定位到 (0, 0)

Z2; 快速下降到离工件上表面 2mm 处

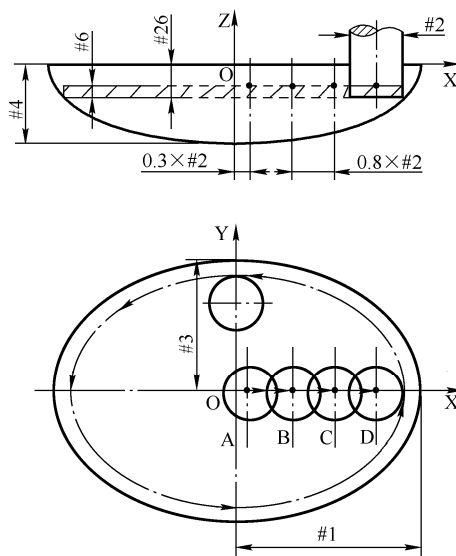


图 6-32 内椭圆球面粗加工

```
#8=0.8*#2; 设定行间距
#18=SQRT[1-[#2*#2/4]/[#3*#3]]*#4; 计算立铣刀最大背吃刀量#18
#19=FIX[#18/#6]; 计算深度切削次数
#14=0; 给深度切削次数计数器变量#14 赋初始值 0
WHILE[#14LE#19] DO3; 循环 3 设定条件: 变量#14 不大于深度切削次数
#26=#6*[#14+1]; 计算每层背吃刀量 Z 轴坐标
IF[#26GT#18] THEN#26=#18; 限定最大背吃刀量#18
G01 Z-#26 F[0.3*#7]; 直线缓慢切削到相应层背吃刀量#26
#30=SQRT[1-#26*#26/[#4*#4]]; 中间计算值寄存在变量#30 中
#20=#30*#1; 计算在深度#26 时, 截面椭圆 X 轴向半轴长
#21=#30*#3; 计算在深度#26 时, 截面椭圆 Y 轴向半轴长
IF[#20GE#21] THEN#9=#20; 确定每层 X、Y 向切削余量中的最大值
IF[#20LT#21] THEN#9=#21; 确定每层 X、Y 向切削余量中的最大值
#10=FIX[#9/#8]; 计算水平切削需要切削的次数
#15=0; 给水平切削次数计数器值变量#15 赋初始值 0
WHILE[#15LE#10] DO2; 循环 2 设定条件: 水平切削次数计数器值不大于
    水平切削需要切削的次数#10
#11=0.3*#2+#8*#15; 计算刀具中心 X 轴半轴长
#31=#20-#2/2; 设定 X 轴刀具中心最大极限尺寸
IF[#11GT#31] THEN#11=#31; 限定 X 轴刀具中心最大极限尺寸
#32=#21-#2/2; 设定 Y 轴刀具中心最大极限尺寸
#12=0.3*#2+#8*#15; 计算刀具中心 Y 轴半轴长
IF[#12GT#32] THEN#12=#32; 限定 Y 轴刀具中心最大极限尺寸
#16=90; 给角度变量#16 赋初始值 90°
WHILE[#16LE450] DO1; 循环 1 设定条件: 角度变量#16 不大于 450°
#24=#11*COS[#16]; 计算刀具中心 X 轴坐标
#25=#12*SIN[#16]; 计算刀具中心 Y 轴坐标
G01 X#24 Y#25 F#7; 直线切削到刚计算的刀具中心 (#24, #25)
#16=#16+#17; 角度变量#16 递增#17
END1; 循环 1 结束
#15=#15+1; 平面椭圆切削次数计数器值累加 1
END2; 循环 2 结束
G00 Z[2-#26]; 抬刀 2mm
X0 Y0; 快速定位到 (0, 0)
#14=#14+1; 切削层计数器值累加 1
```

END3; 循环 3 结束

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

O0164;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S400;

G65 P6164 A35 B16 C25 I15 K1.5 D300 Q10; 调用宏程序 O6164,
赋相关变量初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 球头铣刀

1) 分析: 如图 6-33 所示, 用球头铣刀进行内椭圆球面的精加工, 内椭圆球 X、Y、Z 轴向的半轴长分别为 #1、#3、#4; 球头铣刀半径为 #2, 建立坐标系。

首先计算好刀具加工时球心 X、Y、Z 轴向的半轴长分别为: $\#11=\#1-\#2$ 、 $\#13=\#3-\#2$ 、 $\#14=\#4-\#2$, 开始加工时刀具球心快速定位在 (0, 0, 2-#14), 缓慢切削到深度-#14, 由下而上, 由中心向外, 逐层椭圆面铣削, 从而完成内椭圆球面的加工。

Z 轴坐标初始值为 $\#26=\#14-\#6$, $Z=-\#26$, 计算此时所在高度内椭圆球面截面椭圆平面的 X、Y 轴向半轴长, 逆时针加工此椭圆平面, Z 轴坐标值递减一个深度递增量 #6, 即抬高一个 #6, 再计算此层椭圆平面的 X、Y 轴向半轴长, 加工此面, 进而一层一层向上加工, 直至完成整个球面加工。

2) 变量及说明:

#1=[A]; 椭圆球 X 轴方向半轴长

#2=[B]; 立铣刀直径

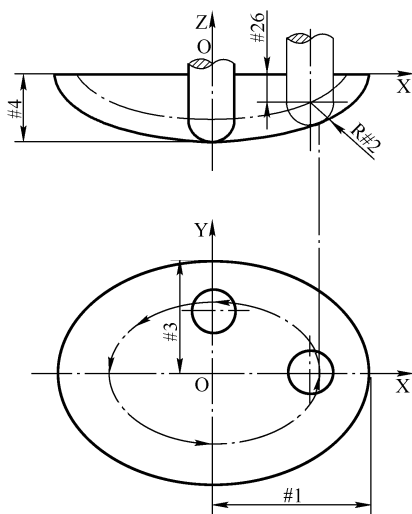


图 6-33 内椭圆球面精加工

#3=[C]; 椭圆球 Y 轴方向半轴长

#4=[I]; 椭圆球 Z 轴方向半轴长

#6=[K]; 深度增量

#7=[D]; 进给量

#17=[Q]; 角度递增量

3) 宏程序如下:

%

O6165; 子程序号

#11=#1-#2; 计算刀具球心 X 轴半轴长

#13=#3-#2; 计算刀具球心 Y 轴半轴长

#14=#4-#2; 计算刀具球心 Z 轴半轴长

G00 X0 Y0; 快速定位到 (0, 0)

Z[-#14]; 快速下降到离工件底面 2mm 处

G01 Z-#14 F[0.3*#7]; 直线缓慢切削到工件底面

#26=#14-#6; 给背吃刀量变量#26 赋初始值#14-#6

WHILE[#26GE0] DO2; 循环 2 设定条件: 背吃刀量变量#26 不小于 0

G01 Z-#26 F#7; 直线切削到背吃刀量#26

#30=SQRT[1-#26*#26/[#14*#14]]; 中间计算值寄存在变量#30 中

#20=#30*#11; 计算刀具球心在相应切削层的 X 轴半轴长

#21=#30*#13; 计算刀具球心在相应切削层的 Y 轴半轴长

X0 Y#21; 直线切削到平面椭圆加工起始点

#16=90; 给角度变量#16 赋初始值 90°

WHILE[#16LE450] DO1; 循环 1 设定条件: 角度变量#16 不大于 450°

#24=#20*COS[#16]; 计算刀具中心 X 轴向坐标

#25=#21*SIN[#16]; 计算刀具中心 Y 轴向坐标

G01 X#24 Y#25; 直线切削到刚计算好的切削点 (#24, #25)

#16=#16+#17; 角度变量#16 累加一个角度递增量#17

END1; 循环 1 结束

#26=#26-#6; 背吃刀量递减一个深度递增量#6

END2; 循环 2 结束

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

```

O0165;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6165 A35 B5 C25 I15 K0.5 D1000 Q10; 调用宏程序 O6165,
                                           赋相关变量初始值

G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.1.7 实例七圆环面加工

1. 立铣刀

1) 分析: 如图 6-34 所示, 圆环面的最大外圆直径#1, 圆环半径#3, 建立坐标系, 用直径#2 的立铣刀进行铣削。

首先刀具定位于圆环右边中心位置 ($[\#1/2+\#2]$, 0), 由下而上, 由外向内, 从圆环面轴线高度 ($Z=-\#3$) 逐层向上顺时针整圆铣削。为了消除切入切出痕迹, 除了采用圆弧切入切出外, 还每层切入切出点沿 XY 平面, 围绕 Z 轴旋转一个角度递增量#6, 并运用 G68 坐标系旋转指令, 简化编程坐标值计算。完成外半圆环加工后, 再快速定位在 ($[\#1/2-\#2]$, 0), 相似方法, 逆时针整圆铣削内半圆环加工。

2) 参数及说明:

#1=[A]; 圆环大圆直径

#2=[B]; 刀具直径

#3=[C]; 圆环面半径

#6=[K]; 角度增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

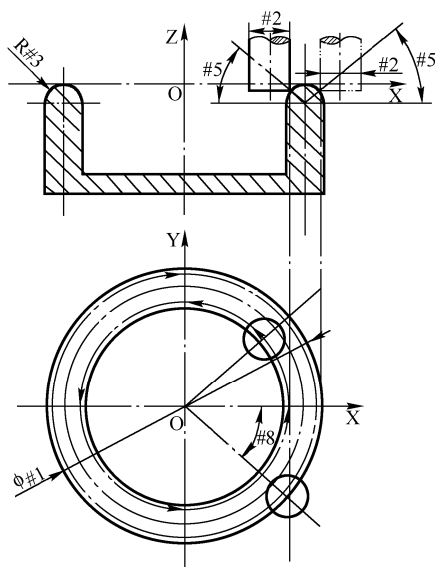


图 6-34 圆环面加工 (1)

%

O6171; 子程序号

G00 X[#1/2+#2] Y0; 快速定位到圆环外上圆弧面加工准备位置

#10=#1/2-#3; 计算圆环中心点 X 轴上坐标

#5=0; 夹角赋初始值为 0

#8=0; 水平旋转角度赋初始值为 0

WHILE[#5LE90] DO1; 循环 1 设定条件: 夹角#5 不大于 90°

G68 X0 Y0 R#8; 坐标系旋转一个角度#8

#24=#10+#3*COS[#5]; 计算相应点的 X 轴坐标

#26=#3*[1-SIN[#5]]; 计算相应点的 Z 轴坐标

G00 Z-#26 F#7; 快速定位于相应点的深度

G41 D1 X[#24+#18] Y#18; 半径补偿到圆弧切入起始点

G03 X#24 Y0 R#18; 圆弧切入

G02 I-#24; 顺时针整圆圆弧切削

G03 X[#24+#18] Y-#18 R#18; 圆弧切出

G40 G0 Y0; 取消半径补偿

#5=#5+#6; 夹角累加一个角度增量#6

#8=#8-#6; 水平旋转角度累减 1 个角度增量#6

G69; 坐标系旋转取消

END1; 循环 1 结束

G00 Z2; 快速抬刀至离工件上平面 2mm 处

X[#1/2-#2] Y#0; 快速定位到圆环内上圆弧面加工准备位置

#5=0; 夹角赋初始值为 0

#8=0; 水平旋转角度赋初始值为 0

WHILE[#5LE90] DO1; 循环 1 设定条件: 夹角#5 不大于 90°

G68 X0 Y0 R#8; 坐标系旋转一个角度#8

#24=#10-#3*COS[#5]; 计算圆环内上圆弧面相应点的 X 轴坐标

#26=#3*[1-SIN[#5]]; 计算圆环内上圆弧面相应点的 Z 轴坐标

G01 Z-#26 F#7; 直线切削到背吃刀量-#26

G41 D1 X[#24-#18] Y-#18; 半径补偿到圆弧切入起始点

G03 X#24 Y0 R#18; 圆弧切入

G03 I-#24; 逆时针整圆圆弧切削

G03 X[#24-#18] Y#18 R#18; 圆弧切出

G40 G0 Y0; 取消半径补偿

#5=#5+#6; 夹角累加一个角度增量#6

#8=#8-#6; 水平旋转角度累减 1 个角度增量#6

G69; 坐标系旋转取消

END1; 循环 1 结束

G00 Z50; 快速上升到 Z=50

M99; 子程序

%

4) 主程序如下:

%

O171;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6171 A70 B16 C5 K5 D1000 R10; 调用宏程序 O6171, 赋相关变量初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 球头铣刀

1) 分析: 如图 6-35 所示, 此处用的刀具是球头铣刀, 加工方法与图 6-34 立铣刀加工基本相同, 只是刀具参考点是球心, 其坐标计算有所不同。

2) 参数及说明:

#1=[A]; 圆环大圆直径

#2=[B]; 球头铣刀半径

#3=[C]; 圆环面半径

#6=[K]; 角度增量

#7=[D]; 进给量

3) 宏程序如下:

%

O6172; 子程序号

G00 X[#1/2+#2+2] Y0; 快速定位到准备位置

Z-#3; 快速下降到背吃刀量-#3

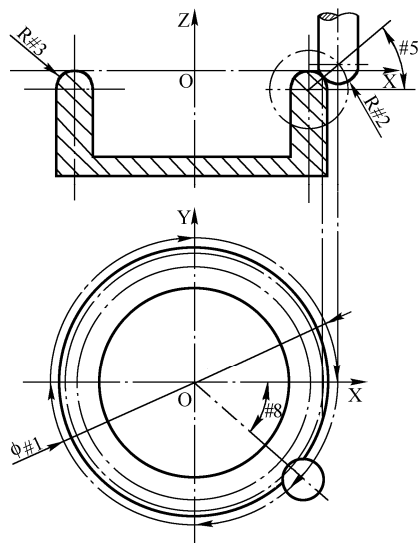


图 6-35 圆环面加工 (2)

```

#10=#1/2-#3; 计算圆环中心点 X 轴上坐标
#5=0; 夹角赋初始值为 0
#8=0; 水平旋转角度赋初始值为 0
WHILE[#5LE90] DO1; 循环 1 设定条件: 夹角#5 不大于 90°
G68 X0 Y0 R#8; 坐标系旋转一个角度#8
#24=#10+[#2+#3]*COS[#5]; 计算铣削圆环外上半圆弧面相应点的 X 坐标
#26=[#2+#3]*SIN[#5]-#3; 计算铣削圆环外上半圆弧面相应点的 Z 坐标
G01 X#24 Y0 Z#26 F#7; 直线切削到对应点
G02 I-#24; 顺时针整圆圆弧切削
#5=#5+#6; 夹角累加一个角度增量#6
#8=#8+#6; 水平旋转角度累加 1 个角度增量#6
G69; 坐标系旋转取消
END1; 循环 1 结束
G00 Z[2+#2]; 抬刀刀尖离工件上平面 2mm
X[#1/2-#2-2] Y0; 快速定位到准备位置
G01 Z-#3; 快速下降到背吃刀量-#3
#5=0; 夹角赋初始值为 0
#8=0; 水平旋转角度赋初始值为 0
WHILE[#5LE90] DO1; 循环 1 设定条件: 夹角#5 不大于 90°
G68 X0 Y0 R#8; 坐标系旋转一个角度#8
#24=#10-[#2+#3]*COS[#5]; 计算铣削圆环内上半圆弧面相应点的 X 坐标
#26=[#2+#3]*SIN[#5]-#3; 计算铣削圆环内上半圆弧面相应点的 Z 坐标
G01 X#24 Y0 Z#26 F#7; 直线切削到对应点
G03 I-#24; 逆时针整圆圆弧切削
#5=#5+#6; 夹角累加一个角度增量#6
#8=#8-#6; 水平旋转角度累减 1 个角度增量#6
G69; 坐标系旋转取消
END1; 循环 1 结束
G00 Z50; 快速上升到 Z=50
M99; 子程序
%
4) 主程序如下:
%
O172;
G54 G90 G00 G43 H1 Z50;

```


X0 Y0;

M03 S1000;

G65 P6172 A70 B8 C5 K5 D1000; 调用宏程序 O6172, 赋相关变量
初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

3. 环形凹面的加工

1) 分析: 如图 6-36 所示, 圆环槽半径 $R\#3$, 环中心线所在圆直径 $\phi\#1$, 准备用 $R\#2$ 球头铣刀进行铣削, 显然 $\#2 < \#3$, 建立坐标系。首先进行粗加工, 球头铣刀定位到圆环槽中心上方及 $(\#1/2, 0)$, 深度余量 $\#3$, 每次背吃刀量 $\#6$, 共需 $\#3/\#6$ 次切削, 每直线切削一个深度 $\#6$, 逆时针圆弧整圆铣削, 一层一层依次进行。完成粗加工后, 设置 X 轴上切削点与 X 轴上圆环槽中心点连线和垂直面之间的夹角为角度变量 $\#5$, 切削点与 Y 轴连线和 X 轴之间的夹角设为角度变量 $\#8$, 精加工开始点为槽与 X 轴交点 (X 坐标值最大), 此时 $\#5=90^\circ$, $\#8=0$, 相应 X、Y、Z 坐标为: $(\#1/2+\#3*\sin[\#5], 0, \#3*\cos[\#5])$, 顺时针整圆铣削环形凹面, 角度变量 $\#5$ 递减一个角度递增量 $\#6$, 旋转角度变量 $\#8$ 递增一个角度递增量 $\#6$, 直到角度变量 $\#5$ 减至 -90° 后铣削完成。

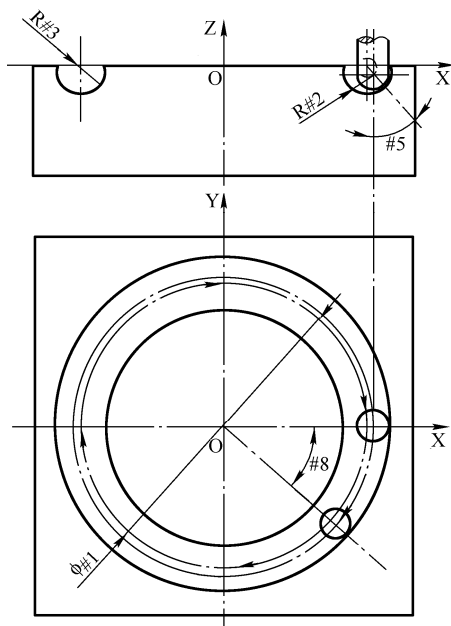


图 6-36 环形凹面的加工

2) 参数及说明:

$\#1=[A]$; 圆环凹面中心所在圆直径

$\#2=[B]$; 球头铣刀半径

$\#3=[C]$; 圆环凹面半径

$\#6=[K]$; 深度增量

$\#7=[D]$; 进给量

$\#17=[Q]$; 角度递增量

3) 宏程序如下:

%

O6173; 子程序号

G00 X[#1/2] Y#0; 快速定位到准备位置
Z[2+#2]; 快速下降到 Z=2+#2 处
#10=FIX[#3/#6]; 先进行圆环中心余量的切除, 计算深度切削次数
#4=0; 给深度计数器赋初始值为 0
WHILE[#4LE#10] DO1; 循环 1 设定条件: 变量#4 不大于#10
#26=#2-#6*[#4+1]; 计算每层背吃刀量#26
IF[#26GT#3] THEN#26=#3; 如果计算出的背吃刀量大于圆环深度, 则限定
它最大为圆环深度
G03 I-[#1/2] Z#26 F#7; 螺旋切削圆环中心余量
#4=#4+1; 计数器计数累加 1
END1; 循环 1 结束
G03 I-[#1/2]; 圆环中心底面平整切削
#5=90; 夹角赋初始值为 90°
#8=0; 水平旋转角度赋初始值为 0
WHILE[#5GE-90] DO1; 循环 1 设定条件: 夹角#5 不小于-90°
G68 X0 Y0 R#8; 坐标系旋转一个角度#8
#24=#1/2+[#3-#2]*SIN[#5]; 计算相应点的 X 轴坐标
#26=-[#3-#2]*COS[#5]; 计算相应点的 Z 轴坐标
G01 X#24 Y0 Z#26; 直线切削到对应点
G02 I-#24; 顺时针整圆圆弧切削
#5=#6-#17; 夹角累减一个角度增量#17
#8=#8+#17; 水平旋转角度累加 1 个角度增量#17
G69; 坐标系旋转取消
END1; 循环 1 结束
G00 Z50; 快速上升到 Z=50
M99; 子程序
%
4) 主程序如下:
%
O173;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6173 A60 B3 C5 I1.5 K10 D1000 Q5; 调用宏程序 O6173,
并赋相关变量初始值

```
G00 G49 Z100;  
X0 Y150;  
M05;  
M30;  
%
```

6.1.8 实例八铣螺纹

1. 外螺纹

现在要用螺纹铣刀加工 $M36 \times 1.5$ 的外螺纹。

1) 分析：通过计算得到螺纹小径： $36\text{mm}-1.3P=34.05\text{mm}$ ；总切削深度： $1.3P=1.3 \times 1.5\text{mm}=1.95\text{mm}$ ；单边背吃刀量分别为 0.675mm 、 0.15mm 、 0.15mm ；对应半径为 17.325mm 、 17.175mm 、 17.025mm 。

2) 变量及说明：

#1=[A]；螺纹大径半径

#2=[B]；螺纹铣刀半径

#3=[C]；螺纹深度

#4=[I]；加工起始高度，取值时考虑其值加螺纹深度能被螺纹导程整除

#5=[J]；螺旋加工时刀具中心所在半径

#7=[D]；进给量

#9=[F]；螺纹导程

3) 宏程序如下：

%

O6181；子程序号

#5=#1+#2；计算螺纹铣刀铣削中心旋转半径

G00 X#5 Y0；快速定位到刀具中心旋转半径处

Z[1+#4]；快速下降到离加工起始高度 1mm 处

G01 Z#4 F[0.3*#7]；直线缓慢切削到起始高度

WHILE[#4GT#3] DO1；循环 1 设定条件：自变量#4 大于螺纹深度值

#4=#4-#9；自变量#4 递减一个螺纹导程值

G02 I-#5 Z#4 F#7；顺时针螺旋铣削螺纹

END1；循环 1 结束

G00 X[#5+10]；快速定位到 $X[\#5+10]$

Z50；快速上升到 $Z=50$

M99；子程序结束

%

4) 主程序如下:

%

O181;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6181 A17.325 B12 C-20 I1 D500 F1.5; 调用宏程序 O6181,
赋相关变量初始值

G65 P6181 A17.175 I1;

G65 P6181 A17.025 I1;

G49 G00 Z100;

X0 Y150;

M05;

M30;

%

2. 内螺纹

现在要用螺纹铣刀加工 $M36 \times 1.5$ 的内螺纹。

1) 分析: 通过计算得到

底孔直径: $36\text{mm} - 1.1P = 34.35\text{mm}$; 总切削深度: $1.1P = 1.1 \times 1.5\text{mm} = 1.65\text{mm}$;
单边背吃刀量分别为 0.5mm 、 0.2mm 、 0.125mm ; 对应半径为 17.75mm 、 17.95mm 、 18.075mm 。

2) 变量及说明:

#1=[A]; 螺纹大径半径

#2=[B]; 螺纹铣刀半径

#3=[C]; 螺纹深度

#4=[I]; 螺纹顶面 Z 轴坐标值 (取绝对值)

#5=[J]; 螺旋加工时刀具中心所在半径

#7=[D]; 进给量

#9=[F]; 螺线程距

3) 宏程序如下:

%

O6182; 子程序号

#5=#1-#2; 计算螺纹铣刀铣削中心旋转半径

#10=FUP[(#3-#4)/#9]; 计算螺纹圈数上取整

G00 X0 Y0; 快速定位到刀具中心旋转半径处

```
Z-[#4+#3]; 快速下降到螺纹深度
G01 Z-[#4+#3+#9/2] F[0.3*#7]; 直线切削到切削准备深度
#8=#3+#4; 设定深度变量初始值
G03 X#5 Z-#8 R[#5/2] F#7; 螺旋切削到螺纹切入点
WHILE[#10GT0] DO1; 循环 1 设定条件: 自变量#8 > 0
#8=#8-#9; 深度变量#8 递减一个螺纹导程值
G03 I-#5 Z-#8 F#7; 逆时针螺旋铣削螺纹
#10=#10-1; 计数器累减 1
END1; 循环 1 结束
G00 X0; 快速定位到 X0
Z50; 快速上升到 Z=50
M99; 子程序结束
%
```

4) 主程序如下:

```
%
O182;
G5490 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6182 A17.75 B12 C-20 I0 D500 F1.5; 调用宏程序 O6182,
                                           赋相关变量初始值

G65 P6182 A17.95 I0;
G65 P6182 A18.075 I0;
G49 G00 Z100;
X0 Y150;
M05;
M30;
%
```

6.2 非圆曲线轮廓零件加工

非圆曲线轮廓零件的种类很多, 但不管是哪一种类型的非圆曲线零件, 编程时所做的数学处理是相同的。一是选择插补方式, 即首先应决定是采用直线段逼近非圆曲线, 还是采用圆弧段逼近非圆曲线; 二是插补节点坐标计算。采用直线段逼近零件轮廓曲线, 一般数学处理较简单, 但计算的坐标数据较多。这里重点

举例说明直线逼近法中的等插补段法编制非圆曲线轮廓零件程序。

等插补段法是使一坐标的增量相等，然后求出曲线上相应的节点，将相邻节点连成直线，用这些直线段组成的折线代替原来的轮廓曲线。其特点是计算简单，坐标增量的选取可大可小，选得越小，加工精度越高，同时节点会增多。

下面通过一些典型的非圆曲线轮廓零件加工编程实例，讲解如何利用宏程序进行此类零件的程序编制。

6.2.1 实例一椭圆

加工如图 6-37 所示椭圆轮廓零件。

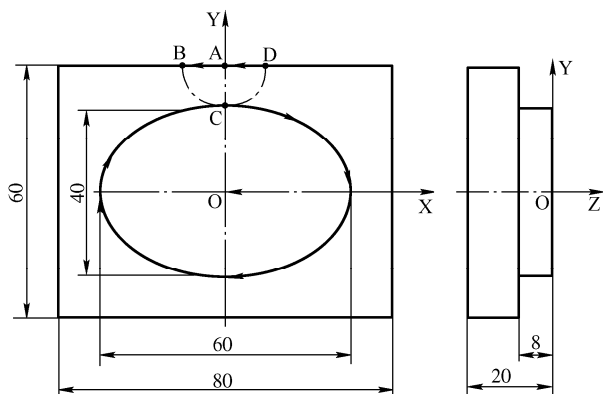


图 6-37 椭圆轮廓零件加工

1) 分析：此零件为方程式 $\frac{x^2}{30^2} + \frac{y^2}{20^2} = 1$ 的椭圆凸台，高度为 8mm，毛坯尺寸为 80mm×60mm×20mm 长方体。利用机用虎钳即可装夹此零件，每次背吃刀量为 1.5mm，需要 8/1.5 次完成深度切削，运用宏程序 FIX[] 函数公式进行深度次数的计算；椭圆轮廓根据参数方程： $X=30\cos\alpha$ ， $Y=20\sin\alpha$ 进行等分点的 X、Y 坐标的计算，其中 α 是任意点与椭圆中心连线与 X 轴正向夹角，角度从 90° 顺时针方向走刀到 -270° 完成整个椭圆的加工；再结合半径补偿指令，从而直接根据轮廓尺寸进行坐标计算、编程，相关变量设定、赋初始值，宏程序见下面所叙。为了具有普遍性，例题以主程序赋初始值调用宏程序。对于不同尺寸的相近零件可以用同一个宏程序，只需赋相应初始值的主程序调用它即可。

2) 变量及说明：

#1=[A]；椭圆 X 轴向半轴长

#2=[B]；椭圆 Y 轴向半轴长

#3=[C]；椭圆厚度

#4=[I]; 切削点与 X 轴正向夹角

#5=[J]; 角度增量

#6=[K]; 背吃刀量增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O621; 子程序号

G00 X0 Y[#18+#2]; 快速定位到加工准备点

Z5; 下降到高度准备点

#10=FIX[#3/#6]; 计算深度加工次数, 上取整

WHILE[#10GE0] DO2; 建立循环 2 设定条件: 变量#10 大于等于 0

#11=#10*#6-#3; 计算每次背吃刀量值, 从后向前总深度递减一个深度增量

G00 G41 D1 X-#18; 建立半径补偿

G03 X0 Y#2 Z#11 R#18 F#7; 1/4 螺旋切入

#4=90; 变量#4 作为椭圆任意切削点和原点连线与 X 轴正向夹角, 赋初始值为 90°

WHILE[#4GE-270] DO1; 建立循环 1 设定条件: 变量#4 大于等于-270°

#24=#1*COS[#4]; 计算变量#24 坐标值

#25=#2*SIN[#4]; 计算变量#25 坐标值

G01 X#24 Y#25; 直线插补 X、Y 坐标分别为变量#24、#25

#4=#4-#5; 变量#4 顺时针递减一个角度增量#5

END1; 循环 1 结束

G03 X#18 Y[#2+#18] Z[2+#11] R#18; 1/4 圆弧螺旋切出

G00 G40 X0; 取消半径补偿

#10=#10-1; 变量#10 (深度切削次数) 递减

END2; 循环 2 结束

G00 Z100; 快速抬高至 Z100 处

M99; 子程序结束

%

4) 主程序如下:

%

O21; 主程序号

G54 G90 G00 G43 H1 Z50; G54、G90; 建立长度补偿, 快速下降到 Z50
X0 Y0; 快速定位至 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P621 A30 B20 C8 J5 K2.5 D300 R10; 调用子程序 O621, 赋
相关变量初始值

G00 G49 Z100; 取消长度补偿, 快速抬高至 Z100

X0 Y150; 快速定位至 X0 Y150

M05; 主轴停止

M30; 主程序结束, 返回起始点

%

6.2.2 实例二双曲线

加工如图 6-38 所示双曲线轮廓零件。

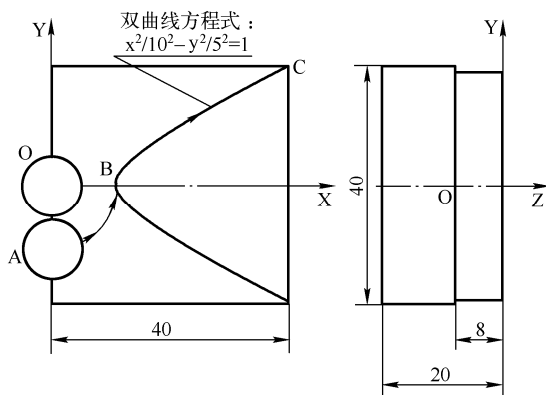


图 6-38 双曲线轮廓加工

1) 分析: 此零件为方程 $\frac{x^2}{10^2} - \frac{y^2}{5^2} = 1$ 的双曲线轮廓凸台, 高度为 8mm, 毛

坯尺寸为 40mm×40mm×20mm 长方体。利用机用虎钳即可装夹此零件, 每次背吃刀量为 1.5mm, 需要 8/1.5 次完成深度切削, 运用宏程序 FIX[] 函数公式进行深度次数的计算; 将轮廓沿 X 轴进行等分, 对应等分点 X 轴的坐标根据方程式进行 Y 轴的坐标计算, 将间距 #4 设为 X 轴坐标值增量变量, 当 X 轴坐标从 #1+#4 开始, 以 #4 为递增量逐步递增时, 算出相应 Y 轴坐标, 得到一系列坐标点, 直线切削拟合这些点即可加工出双曲线轮廓; 再结合半径补偿指令, 从而直接根据轮廓尺寸进行坐标计算、编程。此程序只能加工单边轮廓, 将它设定为宏程序, 通过以 Y 轴为镜像轴的镜像指令调用它, 就可加工出另一边的轮廓。

2) 变量及说明:

#1=[A]; 双曲线实轴长

#2=[B]; 双曲线虚轴长

#3=[C]; 双曲线 X 轴坐标极限值

#4=[I]; X 轴坐标值增量

#5=[J]; 凸台高

#6=[K]; 每次背吃刀量值

#7=[D]; 进给量

3) 宏程序如下:

%

O622; 子程序号

G00 X0 Y0; 快速定位到准备点

Z5; 快速下降到 Z=5

#10=FIX[#5/#6]; 计算深度需要切削次数

WHILE[#10GE0] DO2; 循环 2 设定条件: 深度切削次数不小于 0

#11=#10*#6-#5; 计算每次背吃刀量

G00 G41 D1 Y-#1; 半径补偿

Z[#11+2]; 快速下降到离背吃刀量 2mm 处

G03 X#1 Y0 Z#11 R#1 F#7; 螺旋切入

#24=#1+#4; X 轴向坐标赋初始值

WHILE[#24LE#3] DO1; 循环 2 设定条件: X 轴向坐标值不大于最大极限尺寸

#24=#24+#4; 计算每轮次 X 轴向坐标值

#15=[#24*#24]/[#1*#1]-1; 中间值寄存

#25=SQRT[#15]*#2; 计算每轮次 Y 轴向坐标值

G01 X#24 Y#25; 直线切削到每轮次计算点 (#24, #25)

END1; 循环 1 结束

G00 Z5; 快速抬刀到 Z=5

G00 G40 X0 Y0; 取消半径补偿

#10=#10-1; 深度需要切削次数递减 1

END2; 循环 2 结束

G00 Z50; 快速抬刀到 Z50

M99; 子程序结束

%

4) 主程序如下:

%

O22;

G54 G90 G00 G43 H1 Z50;

```

X0 Y0;
M03 S1000;
G65 P622 A10 B5 C40 I1 J8 K1.5 D300 R10; 调用子程序 O622,
                                           赋相关变量初始值
G51.1 Y0; 建立以 X 轴为镜像轴的镜像坐标系
G65 P622 A10 B5 C40 I1 J8 K1.5 D300 R10; 调用子程序 O622,
                                           赋相关变量初始值
G50.1 Y0; 取消以 X 轴为镜像轴的镜像坐标系
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.2.3 实例三抛物线

加工如图 6-39 所示抛物线轮廓零件。

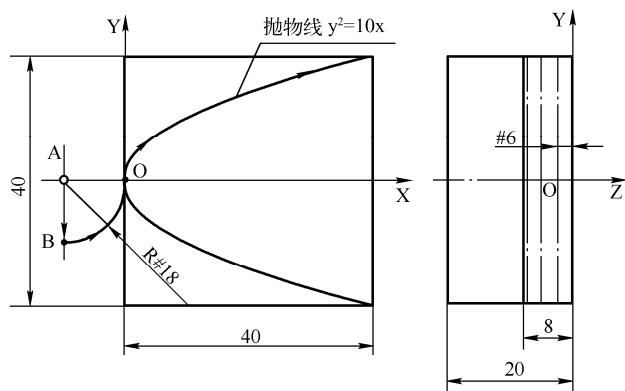


图 6-39 抛物线轮廓加工

1) 分析: 如图 6-39 所示, 此零件为方程式 $y^2=10x$ 的抛物线轮廓凸台, 高度为 8mm, 毛坯尺寸为 40mm×40mm×20mm 长方体。参照图 6-38 分析, 这里不再详述。

2) 变量及说明:

#2=[B]; X 轴向终点坐标值

#3=[C]; 深度

#4=[I]; X 轴向坐标值增量

#6=[K]; 每次背吃刀量值

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O623; 子程序号

G00 X-#18 Y0; 快速定位到圆弧切入准备点

Z5; 快速下降到加工起始高度

#10=FIX[#3/#6]; 计算深度切削次数, 上取整

WHILE[#10GE0] DO2; 循环 2 设定条件: 变量#10 大于等于 0

#11=#10*#6-#3; 计算每层背吃刀量值, 从下往上递增一个#6 值

G00 Z[2+#11]; 快速下刀至距加工平面 2mm 处

G41 D1 Y-#18; 建立半径补偿

G03 X0 Y0 Z#11 R#18 F#7; 螺旋切入

#24=0; 变量#24 赋初始值 0

WHILE[#24LE#2] DO1; 循环 1 设定条件: 变量#24 小于等于 X 轴极限尺寸

#24=#24+#4; 变量#24 递增一个增量#4

#25=SQRT[10*#24]; 计算 X 坐标值对应的 Y 坐标值

G01 X#24 Y#25; 直线插补到计算好的点 (#24, #25)

END1; 循环 1 结束

G00 Z5; 快速上升到加工起始高度

G00 G40 X-#18 Y0; 取消半径补偿

#10=#10-1; 深度加工次数计数器递减 1

END2; 循环 2 结束

G00 Z100; 快速返回到 Z100 高度

M99; 子程序结束

%

4) 主程序如下:

%

O23;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P623 B40 C8 I0.5 K2.5 D300 R10; 调用子程序 O623, 赋相关
变量初始值

G51.1 Y0; 建立以 X 轴为镜像轴的镜像坐标系

G65 P623 B40 C8 I0.5 K2.5 D300 R10; 调用子程序 O623, 赋相关
变量初始值

G50.1 Y0; 取消以 X 轴为镜像轴的镜像坐标系

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

6.2.4 实例四正弦曲线

1. 平面正弦曲线轮廓

加工如图 6-40 所示平面正弦曲线轮廓零件。

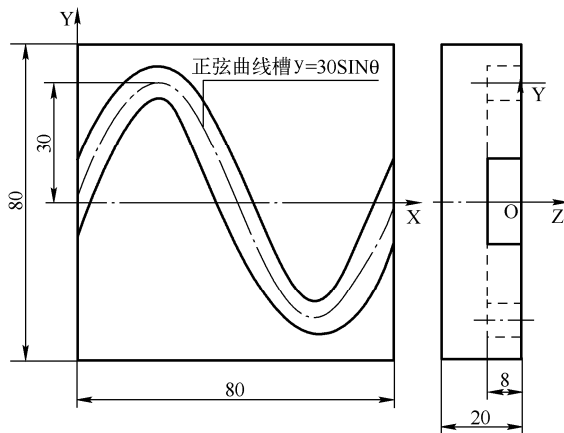


图 6-40 平面正弦曲线轮廓加工

1) 分析: 此零件为方程式 $y=30\text{SIN}\theta$ 的正弦曲线槽, 槽深度为 8mm, 毛坯尺寸为 $80\text{mm} \times 80\text{mm} \times 20\text{mm}$ 长方体。利用机用虎钳即可装夹此零件, 每次背吃刀量为 1.5mm, 需要 $8/1.5$ 次完成深度切削, 运用宏程序 FIX[] 函数公式进行深度次数的计算。从图 6-40 可看出, 在 X 轴向 40mm 距离, 正弦曲线正好是一个周期, 那么单位 X 轴坐标对应的角度为 $360/\#1$, 将轮廓沿 X 轴进行等分, 对应等分点 X 轴的坐标, 计算对应的角度 $\#10$, 根据方程式 $y=30\text{SIN}[\#10]$ 进行 Y 轴的坐标计算, 将间距 $\#4$ 设为 X 轴坐标值增量变量, 当 X 轴坐标从 -5 开始, 以 $\#4$ 为递增量逐步递增时, 算出相应 Y 轴坐标, 得到一系列坐标点, 直线切削拟合这些点即可加工出正弦曲线轮廓。

2) 变量及说明:

#1=[A]; X 轴向工件长度

#2=[B]; 正弦幅值

#4=[I]; X 轴向坐标值增量

#5=[J]; 槽深度

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6241; 子程序号

#11=FIX[#5/#6]; 计算深度需要切削次数

WHILE[#11GE0] DO1; 循环 1 设定条件: 深度切削次数不小于 0

#9=#11*#6-#5; 计算每次背吃刀量

G00 X-5 Y0; 快速定位到 (-5, 0) 处

G01 Z-#9 F[0.3*#7]; 直线切削到每次背吃刀量

#24=-5; 给 X 轴向坐标赋初始值

#20=#1+5; 设定 X 轴向坐标最大极限尺寸

WHILE[#24LE#20] DO2; 循环 2 设定条件: X 坐标值不大于最大极限尺寸

#24=#24+#4; 计算每个等分点 X 轴向坐标值

#10=#24*360/#1; 计算每个等分点对应的角度

#25=#2*SIN[#10]; 计算每个等分点对应的 Y 轴坐标值

G01 X#24 Y#25 F#7; 直线切削到计算好的等分点

END2; 循环 2 结束

G00 Z5; 快速抬刀至 Z=5

#11=#11-1; 深度需要切削次数递减 1

END1; 循环 1 结束

M99; 子程序结束

%

4) 主程序如下:

%

O241;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6241 A80 B30 I1 J8 K1.5 D100 R10; 调用子程序 O6241,
相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 圆柱正弦曲线轮廓

加工如图 6-41 所示圆柱正弦曲线轮廓零件。

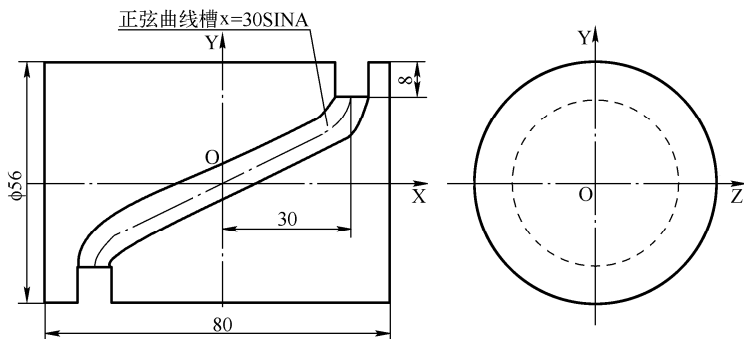


图 6-41 圆柱正弦曲线轮廓加工

1) 分析: 如图 6-41 所示, 基本思路参照图 6-40, 只是多了一个旋转坐标 A, 以旋转坐标 A 角度为变量 #3, #4 为角度增量, 从 #3=0° 到 #3=360°, 计算相应的 X 坐标值, 深度方向借鉴前例, 一系列点用 (X, A) 坐标表示进行拟合加工即可。

2) 变量及说明:

#1=[A]; 圆柱体直径

#2=[B]; 正弦幅值

#3=[C]; 角度

#4=[I]; 角度增量

#5=[J]; 槽深度

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量

3) 宏程序如下:

%

O6242;

#9=#11*#6-#5; 计算每轮次背吃刀量

```
G01 Z[#1/2+#9] F[0.3*#7]; 直线切削下刀到背吃刀量
#3=0; 旋转坐标角度赋初始值为 0
WHILE[#3LE360] DO2; 循环 2 设定条件: 角度不大于 360°
#3=#3+#4; 计算每个等分点对应角度
#24=#2*SIN[#3]; 计算每个等分点对应 X 轴向坐标值
G01 X#24 A#3 F#7; 直线切削到相应坐标点
END2; 循环 2 结束
#11=#11-1; 深度切削次数递减 1
END1; 循环 1 结束
M99; 子程序结束
%
```

4) 主程序如下:

```
%
O242;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
A0;
M03 S1000;
G65 P6242 A56 B30 I3 J8 K1.5 D100; 调用子程序 O6242, 赋相关
                                     变量初始值

G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.2.5 实例五余弦曲线

加工如图 6-42 所示平面余弦曲线轮廓零件。

1) 分析: 参照图 6-40 正弦曲线轮廓加工即可。

2) 变量及说明:

#1=[A]; X 轴向工件长度

#2=[B]; 余弦幅值

#4=[I]; X 轴向坐标值增量

#5=[J]; 槽深度

#6=[K]; 每次背吃刀量增量

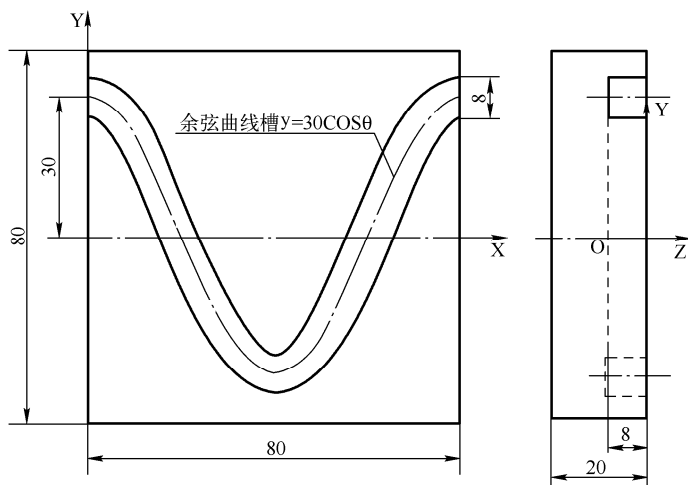


图 6-42 平面余弦曲线轮廓加工

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6251; 子程序号

#11=FIX[#5/#6]; 计算深度需要切削次数

WHILE[#11GE0] DO1; 循环 1 设定条件: 深度切削次数不小于 0

#9=#11*#6-#5; 计算每次背吃刀量

G00 X0 Y#2; 快速定位到 (0, #2) 处

G01 Z#9 F[0.3*#7]; 直线切削到每次背吃刀量

#24=0; 给 X 轴向坐标赋初始值

#20=#1+5; 设定 X 轴向坐标最大极限尺寸

WHILE[#24LE#1] DO2; 循环 2 设定条件: X 坐标值不大于最大极限尺寸

#24=#24+#4; 计算每个等分点 X 轴向坐标值

#10=#24*360/#1; 计算每个等分点对应的角度

#25=#2*COS[#10]; 计算每个等分点对应的 Y 坐标值

G01 X#24 Y#25 F#7; 直线切削到计算好的等分点

END2; 循环 2 结束

G00 Z5; 快速抬刀至 Z=5

#11=#11-1; 深度需要切削次数递减 1

END1; 循环 1 结束

M99; 子程序结束

%

4) 主程序如下:

%

O251;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6251 A80 B30 I1 J8 K1.5 D100 R10; 调用子程序 O6251,
赋相关变量初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

6.2.6 实例六正切曲线

加工图 6-43 所示平面正切曲线轮廓零件。

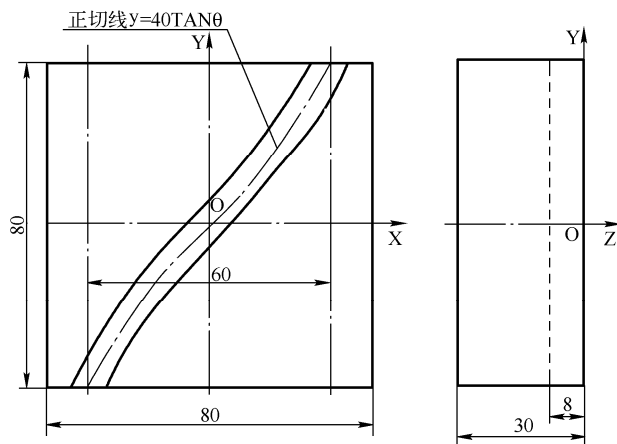


图 6-43 正切曲线轮廓加工

1) 分析: 借鉴正、余弦曲线轮廓加工。

2) 变量及说明:

#1=[A]; 正切线 X 轴向范围值

#2=[B]; 正切线 Y 轴向最大值

#4=[I]; X 轴向坐标值增量

#5=[J]; 槽深

#6=[K]; 每次背吃刀量值

#7=[D]; 进给量

#8=[E]; 正切线系数

3) 宏程序如下:

%

O626; 子程序号

#20=ATAN[#2]/[#8]; 计算最大 X 轴向尺寸对应角度值

G00 X[-#1/2] Y-#2; 快速定位到加工起始点

Z5; 快速下降到 Z=5

#10=FIX[#5/#6]; 计算深度需要切削次数

WHILE[#10GE0] DO1; 循环 1 设定条件: 深度切削次数不小于 0

#11=#10*#6-#5; 计算每轮次背吃刀量

G01 Z#11 F[0.3*#7]; 直线切削到相应深度

#14=-#20; 给角度变量赋初始值

WHILE[#14LE#20] DO2; 循环 2 设定条件: 角度变量不大于最大角度

#14=#14+#4; 每轮次角度计算

#24=#14*0.5*#1/#20; 计算相应 X 轴坐标值

#25=#8*TAN[#14]; 计算相应 Y 轴坐标值

G01 X#24 Y#25 F#7; 直线切削对应点

END2; 循环 2 结束

G00 Z5; 快速抬刀到 Z=5

G00 X[-#1/2] Y-#2; 返回加工起始点

#10=#10-1; 深度切削次数递减 1

END1; 循环 1 结束

G00 Z50; 快速抬刀到 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O26;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P626 A60 B40 I1 J8 K1.5 D100 E40; 调用子程序 O626, 赋
相关变量初始值

G00 G49 Z100;
X0 Y150;
M05;
M30;
%

6.2.7 实例七等速螺线

在机械传动中，常常需要把旋转运动变成直线运动。凸轮装置就是借助凸轮绕定轴旋转推动从动杆上、下往复直线运动。如果需要从动杆作随主动件凸轮等速旋转运动而成一定比例径向增大或减少，此时的平面凸轮轮廓就是等速螺线或阿基米德螺线。所谓的等速螺线或阿基米德螺线是指：从定点 O 出发的射线 L，绕 O 点作等角速度的转动，同时射线上的点 M 从初始点开始沿射线作与等角速度成定比的等速直线运动。它的极坐标方程：

$\rho = \rho_0 + (v/\omega) \theta$ ，其中 ρ_0 为等速螺线初始极半径； ρ 为等速螺线终止极半径； v/ω 为比例系数； θ 为对应极角。

加工如图 6-44 所示等速螺线凸轮轮廓零件。

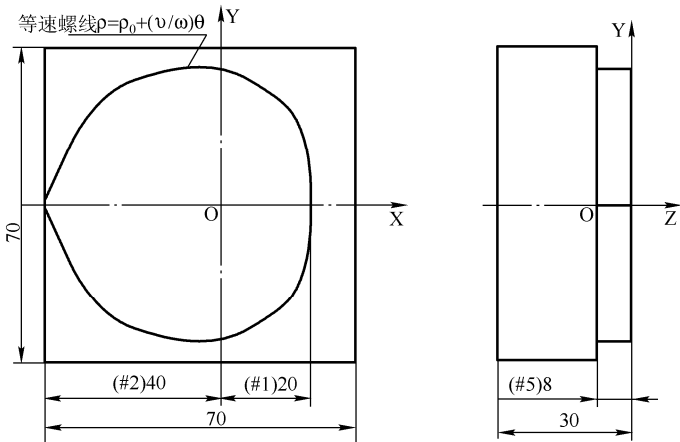


图 6-44 等速螺线凸轮轮廓加工

1) 分析：如图 6-44 所示，要在一工件 70mm×70mm×30mm 的毛坯上加工等速螺线凸轮轮廓。

从工件最右端圆弧切入，顺铣走刀（刀具沿顺时针方向切削），以切削点相对 X 轴正向夹角为角度变量，#8 为角度增量，从 0 递减到 -180°，再从 -180°递减到

-360°, 分别得到 X、Y 轴坐标算术表达式, 系统根据初始值、变化规律和表达式可以得到一系列点的坐标, 通过这些点直线拟合切削, 就可加工此类零件的轮廓。运用半径补偿指令, 直接用零件轮廓相关尺寸编程可简化编程计算。

2) 参数及说明:

#1=[A]; 等速螺线初始极半径

#2=[B]; 等速螺线终止极半径

#3=[C]; 初始角度

#4=[I]; 终止角度

#5=[J]; 凸台高度

#6=[K]; 每次背吃刀量值

#7=[D]; 进给量

#8=[E]; 角度增量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O627; 子程序号

#20=2*[#2-#1]/#4; 计算相应角度

G00 X[#1+#18] Y0; 快速定位到准备点

Z5; 快速下降到 Z=5

#10=FIX[#5/#6]; 计算深度需要切削次数

#11=0; 给深度切削计数器归零

WHILE[#11LE#10] DO1; 循环 1 设定条件: 计数器值不大于深度需要次数

G41 D1 Y#18; 半径补偿

#12=[#11+1]*#6; 计算每轮次背吃刀量

IF[#12GT#5] THEN#12=#5; 限定背吃刀量最大为凸台高度

G03 X#1 Y0 Z-#12 R#18 F#7; 螺旋切入

WHILE[#3GT-#4/2] DO2; 循环 2 设定条件: 角度变量#3 大于设定的终止角

#3=#3-#8; 计算对应角度

#13=#1+#20*ABS[#3]; 计算对应极半径

G16; 建立极坐标

G01 X#13 Y#3; 直线切削到计算的极点

END2; 循环 2 结束

#3=180; 给角度变量赋初始值 180°

WHILE[#3GT0] DO3; 循环 3 设定条件: 角度变量#3 大于 0

#3=#3-#8; 计算相应角度

```
#13=#1+#20*#3; 计算对应极半径
G01 X#13 Y#3; 直线切削到对应点
END3; 循环 3 结束
G15; 极坐标取消
G03 X[#1+#18] Y-#18 Z[-#12+2] R#18; 螺旋切出, 刀具抬高 2mm
G40 G00 Y0; 取消半径补偿
#11=#11+1; 深度切削次数递减 1
END1; 循环 1 结束
M99; 子程序结束
%
4) 主程序如下:
%
O27;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P627 A20 B40 C0 I360 J8 K1.5 D300 E5 R10; 调用子程序 O627,
                                                赋相关变量初始值
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.3 孔系加工

孔系加工在数控铣床上的应用非常普遍, 所以数控铣编程指令专门有孔加工固定循环指令, 但一般配 FANUC 系统的数控铣、加工中心机床只能对单一孔或数量少的孔进行加工时, 直接运用孔加工固定循环指令即可, 当遇到多孔加工时, 每个孔中心坐标的计算将增加编程人员的工作量以及计算难度。而通常这些孔都是按照一定的规律分布的, 这时通过分析找出孔分布的规律, 利用宏程序将烦琐的计算交给系统进行运算, 下面例举几个典型的实例讲解宏程序在这方面的应用。

6.3.1 实例一圆周钻孔

加工如图 6-45 所示圆周均布孔, 孔数是 9 个, 均匀分布在直径为 40mm 的圆

周上, 圆周中心坐标 (10, 10), 其中第一孔中心
和圆周中心连线与 X 轴正向夹角 34° , 孔深 15mm。

加工分析:

各个孔加工主要是孔中心坐标值的计算, 如果将圆周中心设为局部坐标原点, 各个孔中心坐标就是圆周半径乘以夹角正、余弦, 孔间角计算 $360^\circ/\text{孔数}$ 。设定局部坐标原点可用 G52 X#8 Y#9, 其中#8、#9 分别是圆周中心 X、Y 坐标 (10, 10); 各孔坐标等于 $R \times \cos[34^\circ + \text{孔间角} \times (\text{序数} - 1)]$ 、 $R \times \sin[34^\circ + \text{孔间角} \times (\text{序数} - 1)]$ 。

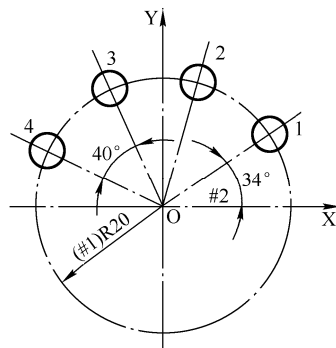


图 6-45 圆周均布孔加工

1. 孔中心直角坐标值计算法编程

1) 分析: 运用上述分析, 每个孔的 X、Y 坐标值分别为:

#24=#1*COS[#5]

#25=#1*SIN[#5]

2) 变量及说明:

#1=[A]; 圆周半径#1=R

#2=[B]; 第一个孔中心与 X 轴正向夹角#2

#3=[C]; 孔数#3=n

#5=[J]; 第 n 个孔的中心与 X 轴正向夹角

#7=[D]; 进给量, 单位为 mm/min

#18=[R]; R 平面 Z 坐标值

#26=[Z]; 孔的深度

3) 宏程序如下:

%

O6311; 程序号

#10=360/#3; 计算孔间距角

#5=#2; 给变量#5 赋值: 第一个孔的中心与 X 轴正向夹角

N5 #24=#1*COS[#5]; 第 n 个孔的 X 轴坐标值

#25=#1*SIN[#5]; 第 n 个孔的 Y 轴坐标值

G81 G99 X#24 Y#25 Z-#26 R#4 F#7; 钻孔循环指令

#3=#3-1; 孔数递减值作为判定条件

#5=#5+#10; 第 n 个孔的中心与 X 轴正向夹角计算

IF[#3GT0] GOTO5; 当孔数递减值大于 0 时, 跳到顺序号为 5 的程序

M99; 子程序结束

%

4) 主程序如下:

%

O311;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6311 A30 B34 C9 R5 D300 Z15; 调用宏程序 O6311, 给相关
变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 孔中心极坐标值计算法编程

1) 分析: 如果用极坐标表示每个孔的中心坐标, 则

G16;

G81 G99 X#1 Y#2 Z-#26 R#18 F#7; 第一孔中心坐标 (#1, #2)

G91 Y#10 K[#3-1]; 其余孔中心坐标极角增量#10=360/#3, 数量为#3-1,
因为第一孔已经加工好

G90;

G15;

2) 变量及说明:

同孔中心直角坐标值计算法编程。

3) 宏程序如下:

%

O6312; 子程序号

#10=360/#3; 计算孔间距角

G16; 建立极坐标

G81 G99 X#1 Y#2 Z-#26 R#18 F#7; 钻孔循环指令

G91 Y#18 K[#3-1]; 增量编程, 以孔间距角为增量, 孔数为#3-1

G90; 恢复绝对值编程

G15; 取消极坐标

M99; 子程序结束

%

4) 主程序: 同主程序 O311 调用。

6.3.2 实例二定角度均布孔

加工如图 6-46 所示定角度圆周均布孔，孔数是 9 个，均匀分布在直径为 40mm 的夹角 120° 圆周上，圆周中心坐标 (10, 10)，其中第一孔中心和圆周中心连线与 X 轴正向夹角 34° ，孔深 15mm。

1. 孔中心直角坐标值算法编程

1) 分析：思路与上节圆周均布孔相似，只是孔间角计算不是 $360^\circ/\text{孔数}$ ，而是定角度 $120^\circ/(\text{孔数}-1)$ 。

2) 变量及说明：

#1=[A]；圆周半径#1=R

#2=[B]；第一个孔中心与 X 轴正向夹角#2

#3=[C]；孔数#3=n

#5=[J]；第 n 个孔的中心与 X 轴正向夹角

#6=[K]；角度增量

#7=[D]；进给量，单位为 mm/min

#18=[R]；R 平面 Z 坐标值

#26=[Z]；孔的深度

3) 宏程序如下：

%

O6321；子程序号

#10=#6/[#3-1]；计算孔间距角

#5=#2；给变量#5 赋值：第一个孔的中心与 X 轴正向夹角

N5#24=#1*COS[#5]；第 n 个孔的 X 轴坐标值

#25=#1*SIN[#5]；第 n 个孔的 Y 轴坐标值

G81 G99 X#24 Y#25 Z-#26 R#18 F#7；钻孔循环指令

#5=#5+#10；第 n 个孔的中心与 X 轴正向夹角计算

#3=#3-1；未加工孔数作为判定条件

IF[#3GT0] GOTO6；当未加工孔数大于 0 时，跳到序号 5 的程序

M99；子程序结束

%

4) 主程序如下：

%

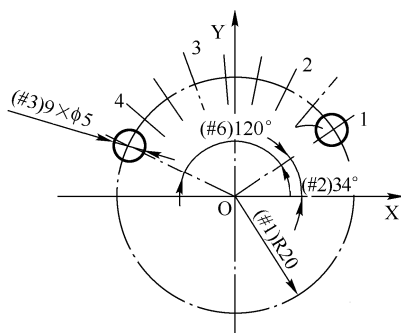


图 6-46 定角度圆周均布孔加工


```
O321;  
G54 G90 G00 G43 H1 Z50;  
X0 Y0;  
M03 S1000;  
G65 P6321 A20 B34 C9 R5 K120 D300 Z15; 调用宏程序 O6321,  
                                           给相关变量赋初始值  
  
G00 G49 Z100;  
X0 Y150;  
M05;  
M30;  
%
```

2. 孔中心极坐标值算法编程

1) 分析: 略。

2) 变量及说明: 同 1。

3) 宏程序如下:

```
%  
O6322; 子程序号  
#10=#6/[#3-1]; 计算孔间距角  
G16; 建立极坐标  
G81 G99 X#1 Y#2 Z-#26 R#18 F#7; 钻孔循环指令  
G91 Y#10 K[#3-1]; 增量编程, 以孔间距角为增量, 孔数为#3-1  
G90; 恢复绝对值编程  
G15; 取消极坐标  
M99; 子程序结束  
%
```

6.3.3 实例三矩阵孔加工

1. 沿直线均布孔加工

加工如图 6-47 所示沿直线均布孔, 孔数是 7 个, 均匀分布在与水平夹角为#2 的直线上, 孔间距为#1, 孔深 15mm。

(1) 绝对值编程

1) 分析: 如图 6-47 所示, 孔的分布规律是沿与 X 轴正向夹角为#2 的直线间距为#1 线性均匀, 将左边第一孔中心坐标设定为坐标原点(可用 G52 X#24 Y#25 指令设定), 利用坐标旋转指令 G68 X0 Y0 R#2, 各孔坐标就简化了, 只有 X 轴坐标需要计算, Y 轴坐标都为 0。

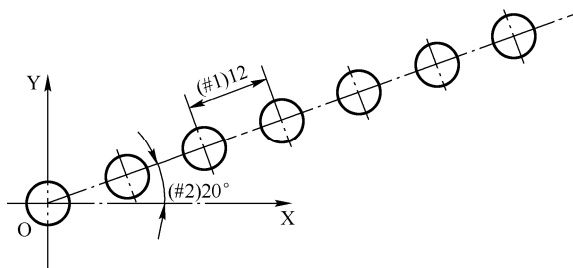


图 6-47 沿直线均布孔加工

借助宏程序中条件跳转指令，设计数器变量#5 为自变量，当#5 小于给定孔数#3 时，返回孔心距计算公式，算出相应孔心 X 轴坐标，接着调用孔循环指令进行该孔的加工。

2) 变量及说明：

#1=[A]；孔间距

#2=[B]；均布线与 X 轴正向夹角#2

#3=[C]；孔数#3=n

#7=[D]；进给量，单位为 mm/min

#18=[R]；R 平面 Z 坐标值

#26=[Z]；孔的深度

3) 宏程序如下：

%

O6331；子程序号

G68 X0 Y0 R#2；以局部坐标系原点为中心，坐标系旋转#2

#5=0；已加工孔数赋零

N5 #10=#1*#5；第 n 个孔的中心 X 轴坐标值计算

G81 G99 X#10 Y0 Z-#26 R#18 F#7；钻孔循环指令

#5=#5+1；计数递增

IF[#5LT#3] GOTO5；当已加工孔数小于 1 时，跳到顺序号 5 的程序

G69；坐标系旋转取消

M99；子程序结束

%

4) 主程序如下：

%

O331；

G54 G90 G00 G43 H1 Z50；

X0 Y0；

M03 S1000;

G65 P6331 A20 B34 C7 R5 D300 Z15; 调用宏程序 O6331, 给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 增量值编程

1) 分析: 在前例分析的基础上, 利用孔循环指令中的重复功能 K, 先绝对值编程进行孔循环指令加工: G81 G99 X0 Y0 Z-#26 R#18 F#7, 接着设定为增量编程, 孔间距作为 X 轴坐标值增量, K 重复次数为“孔数-1”, 该程序段为: G91 X#1 K[#3-1], 这里 K 值为何是“孔数-1”? 因为上一条程序段已经加工了第一个孔, 所以它要减去 1。

2) 变量及说明:

#1=[A]; 孔间距

#2=[B]; 均布线与 X 轴正向夹角

#3=[C]; 孔数

#7=[D]; 进给量, 单位为 mm/min

#18=[R]; R 平面 Z 坐标值

#26=[Z]; 孔的深度

3) 宏程序如下:

%

O6332; 子程序号

G68 X0 Y0 R#2; 以局部坐标系原点为中心, 坐标系旋转#2

G81 G99 X0 Y0 Z-#26 R#18 F#7; 钻孔循环指令

G91 X#1 K[#3-1]; 以孔间距为 X 坐标增量加工#3-1 孔

G90 G69; 绝对值编程坐标系旋转取消

M99; 子程序结束

%

(3) 极坐标编程

1) 分析: 在前面分析的基础上, 下面采用的是极坐标指令进行编程, 再运用孔循环指令中的重复功能, 不用坐标旋转指令。

2) 变量及说明:

#1=[A]; 孔间距

#2=[B]; 均布线与 X 轴正向夹角
 #3=[C]; 孔数
 #7=[D]; 进给量, 单位为 mm/min
 #18=[R]; R 平面 Z 坐标值
 #26=[Z]; 孔的深度
 3) 宏程序如下:

%

O6333; 子程序号

G81 G99 X0 Y0 Z-#26 R#18 F#7; 钻孔循环指令

G16; 极坐标建立

X#1 Y#2; 孔间距为极半径, 夹角为极角

G91 X#1 K[#3-2]; 以孔间距为 X 坐标增量加工 #3-2 孔

G90 G15; 绝对值编程坐标系, 极坐标取消

M99; 子程序结束

%

2. 矩阵分布孔加工

加工如图 6-48 所示矩阵分布孔, 行数是 3, 列数是 5, 行间距是 20, 列间距是 15, 每行孔中心连线与 X 轴夹角是 15° , 每列孔中心连线与 Y 轴夹角是 30° , 底行左边第一个孔中心坐标 (#8, #9), 孔深 15mm。

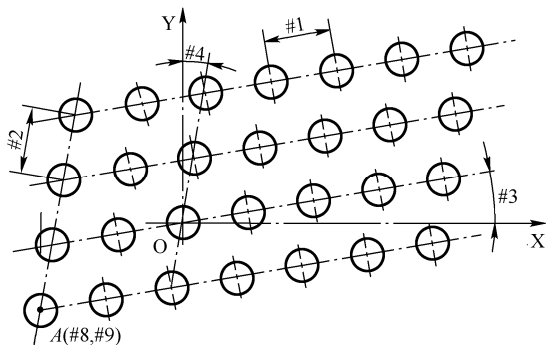


图 6-48 矩阵分布孔加工

1) 分析: 如图 6-48 所示, 孔成矩阵分布, 为了简化坐标值计算, 首先利用局部坐标系指令 (G52 X#8 Y#9) 将左下角第一个孔中心设定为局部坐标系原点, 再运用坐标旋转指令: G68 X0 Y0 R#3, Y 轴坐标计算就简单了, 底下第一行 Y 轴坐标为 0, 向上一行 Y 轴坐标为 $\#2 \times \cos[\#3+\#4]$, 依此类推每增加一行, Y 轴坐标值 $+\#2 \times \cos[\#3+\#4]$ 。

X 轴坐标: 底下第一行从左往右, 第一孔中心坐标为 0, 第二个孔中心坐标为 #1, 再向右每增加一个孔, 坐标值 $+\#1$ 。向上第二行左边第一个孔中心坐标 $\#2 \times \sin[\#3+\#4]$, 向右每增加一个孔, 坐标值 $+\#1$ 。

2) 变量及说明:

#1=[A]; 列间距

#2=[B]; 行间距

#3=[C]; 每行孔中心连线与 X 轴间的夹角

#4=[I]; 每列孔中心连线与 Y 轴间的夹角

#5=[J]; 行数

#6=[K]; 列数

#7=[D]; 进给量, 单位为 mm/min

#18=[R]; R 平面坐标

#26=[Z]; 孔的深度

3) 宏程序如下:

%

O6334; 子程序号

G52 X#8 Y#9; 以底行左边第一个孔中心为局部坐标系原点

G68 X0 Y0 R#3; 以局部坐标系原点为中心, 坐标系旋转#3

#10=0; 给行数计数器赋初始值为 0

WHILE[#10LT#5] DO1; 循环 1 设定条件: 行数计数器值小于给定行数

#11=0; 给列数计数器赋初始值为 0

WHILE[#11LT#6] DO2; 循环 2 设定条件: 列数计数器值小于给定列数

#24=#1*#11+#2*SIN[#4+#3]*#10; 计算任意孔的中心 X 轴坐标值

#25=#2*COS[#4+#3]*#10; 计算任意孔的中心 Y 轴坐标值

G98 G81 X#24 Y#25 Z-#26 R#18 F#7; 孔加工指令

#11=#11+1; 行数累加 1

END2; 循环 2 结束

#10=#10+1; 列数累加 1

END1; 循环 1 结束

G80 G69; 孔加工固定循环及坐标系旋转取消

G52 X0 Y0; 局部坐标系取消, 恢复 G54 坐标系

M99; 子程序结束

%

4) 主程序如下:

%

O334;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

#8=-30;

#9=-30;

G65 P6334 A20 B15 C15 I30 J3 K5 D300 R5 Z15; 调用宏程序 O6331,

给相关变量赋初始值

```
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.3.4 实例四三角均布孔加工

1. 水平对称分布孔加工

加工如图 6-49 所示水平对称分布孔，行数是 3 (#3)，底行是 6 (#4) 个孔，向上依次递减一个孔，水平孔间距是 10 (#1)，列间距是 20 (#2)，以 Y 轴为对称轴均匀分布。底行左边第一个孔中心坐标 (#8, #9)，孔深 15mm。

1) 分析：如图 6-49 所示，利用局部坐标系指令 G52 X#8 Y#9，将底下一行孔中心 X 轴对称点设定为局部坐标系原点，每行孔数计算：最底下一行是 #4，向上依次递减 $\#4=\#4-1$ ，每行最左边的孔中心 X 轴坐标： $-\#1*[\#4-1]/2$ ，再运用孔循环指令中的重复功能 K[#4-1]，就很方便将此行孔逐一加工出来。Y 轴坐标：最底下一行为 0，向上逐一加 #2。

2) 变量及说明：

#1=[A]；水平间距

#2=[B]；垂直间距

#3=[C]；行数

#4=[I]；一行最多孔数

#7=[D]；进给量，单位为 mm/min

#18=[R]；R 平面 Z 坐标值

#26=[Z]；孔的深度

3) 宏程序如下：

%

O6341；子程序号

G52 X#8 Y#9；以底行左边第一个孔中心为局部坐标系原点

#11=0；设定变量 11 为行数计数器并赋初始值为 0

WHILE[#11LT#3] DO1；循环 1 设定条件：行数计数器值小于总行数

#24=#1*#4-1]/2；第 n 个孔的中心 X 轴坐标值计算

#25=#2*#11；第 n 个孔的中心 Y 轴坐标值计算

G90 G81 G99 X-#24 Y#25 Z-#26 R#18 F#7；孔加工循环指令

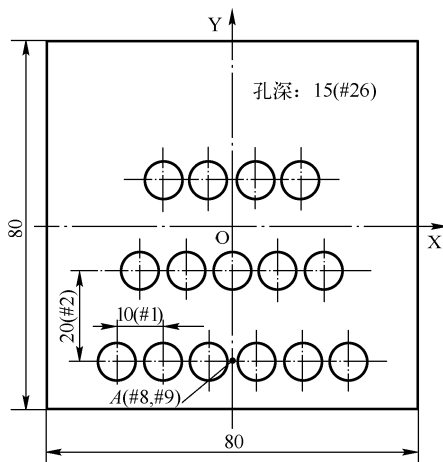


图 6-49 水平对称分布孔加工

G91 X#1 K[#4-1]; 增量方式, X 轴向间距#1, 孔数#4-1

G90; 恢复绝对值编程

#4=#4-1; 孔数递减 1

#11=#11+1; 行数递增 1

END1; 循环 1 结束

G52 X0 Y0; 取消局部坐标系, 恢复 G54 坐标系

M99; 子程序结束

%

4) 主程序如下:

%

O341;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

#8=-25;

#9=-30;

G65 P6341 A10 B20 C3 I6 J5 D300 Z15; 调用宏程序 O6341, 给
相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 左侧对齐均布孔加工

加工如图 6-50 所示均布孔, 行数是 3, 底行是 4 个孔, 向上依次递减一个孔, 水平孔间距是 10 (#1), 列间距是 20 (#2), 以 Y 轴为对称轴均匀分布。底行左边第一个孔中心坐标 (#8, #9), 孔深 15mm。

1) 分析: 如图 6-50 所示, 利用局部坐标系指令 G52 X#8 Y#9, 将左下角第一个孔中心设定为局部坐标系原点, 相应的此孔中心坐标为 (0, 0), 底下一行孔中心的坐标: Y 轴坐标都

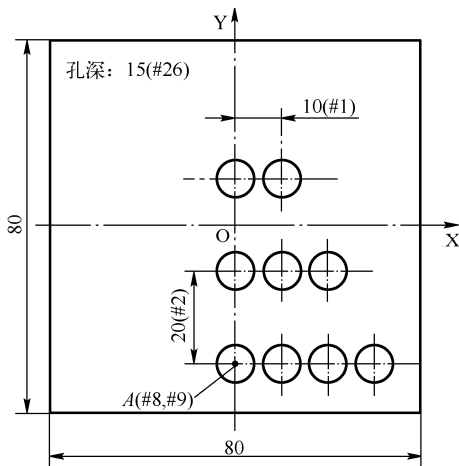


图 6-50 左侧对齐均布孔加工

是0，X轴坐标从左往右第一个孔为0，往后坐标递增量就是#1。

向上第二行，X轴坐标与第一行坐标计算相同，只是少一个孔，Y轴坐标都是#2；再向上就是Y轴坐标增加一个#2。

2) 变量说明：

#1=[A]；水平间距

#2=[B]；垂直间距

#3=[C]；行数

#4=[I]；一行最多孔数

#5=[J]；R平面

#7=[D]；进给量，单位为 mm/min

#26=[Z]；孔的深度

3) 宏程序如下：

%

O6342；子程序号

G52 X#8 Y#9；以底行左边第一个孔中心为局部坐标系原点

#11=0；设定变量11为行数计数器并赋初始值为0

WHILE[#11LT#3] DO1；循环1设定条件：行数计数器值小于总行数

#25=#2*#11；第n个孔的中心X轴坐标值计算

G90 G81 G99 X0 Y#25 Z-#26 R#18 F#7；孔加工循环指令

G91 X#1 K[#4-1]；增量方式，X轴向间距#1，孔数#4-1

G90；恢复绝对值编程

#4=#4-1；孔数递减1

#11=#11+1；行数递增1

END1；循环结束

G52 X0 Y0；取消局部坐标系，恢复G54坐标系

M99；子程序结束

%

4) 主程序如下：

%

O342；

G54 G90 G00 G43 H1 Z50；

X0 Y0；

M03 S1000；

#8=-25；

#9=-30；

G65 P6342 A10 B20 C3 I6 J5 D300 Z15; 调用宏程序 O6342, 给
相关变量赋初始值

```
G00 G49 Z100;  
X0 Y150;  
M05;  
M30;  
%
```

6.3.5 实例五可变深孔加工

1. 等差数列深孔加工

如图 6-51 所示, 钻孔采用多次钻削, 第一孔钻深#1, 所有钻深成等差数列(系数为#6)递减变化规律, 最小钻深#2, 参照深孔循环指令 G73 和 G83 加工形式编写宏程序。

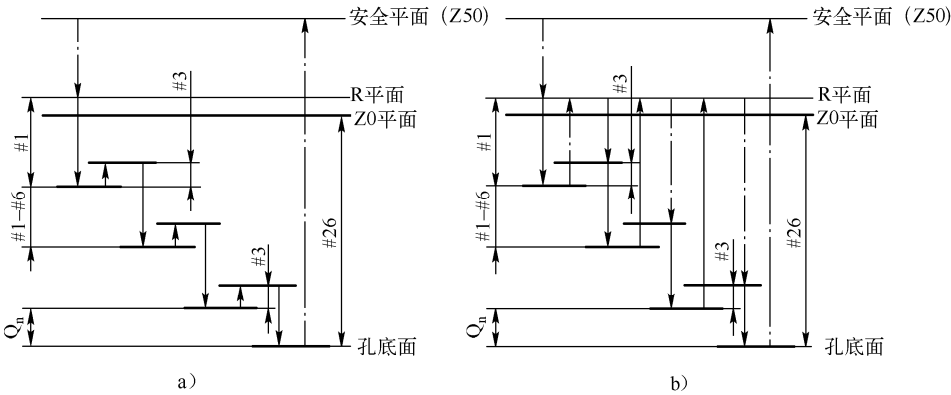


图 6-51 等差数列深孔加工

a) G73 b) G83

1) 分析: 如图 6-51a 所示, 孔加工切削流程: 水平 XY 位置快速定位, 快速下刀至 R 平面, 从 R 平面开始第一次钻削, 深度为#1, 但 Z 轴坐标值不是-#1, 而是-#1+#18, 快速抬刀#3, 接着进行第二次钻削, 下深距离为#1-#6 (#6 为等比系数), Z 轴坐标值为-#1+#18-#1+#6, 再抬刀#3, 接着进行第三次钻削, 下深距离为#1-#6*2, 后面依此类推。

2) 变量及说明:

- #1=[A]; 第一次钻深
- #2=[B]; 最小钻深
- #3=[C]; 回退量
- #6=[K]; 递减量

#7=[D]; 进给量, 单位为 mm/min

#18=[R]; R 平面

#26=[Z]; 孔的深度

3) 宏程序如下:

① G73 钻孔方式。

O6351; 子程序号

G00 Z#18; 快速下降到 R 平面

#10=#18+#26; 总钻削深度 (绝对值)

#8=#1; 给每次钻削深度赋初始值: 第一次钻削深度#1

#9=#10-#8; 第一次钻削后深度剩余量

#11=#18-#8; 第一次钻削深度 Z 坐标值

WHILE[#9GT0] DO1; 循环 1 设定条件: 钻削深度剩余量大于 0

G01 Z#11 F#7; 直线切削到钻削深度

IF[#11EQ-#26] GOTO10; 如果钻削深度等于孔深, 跳转到顺序号 10 程序段

G00 Z[#3+#11]; 刀具退后一个回退量#3

#8=#8-#6; 下一次钻削深度计算

IF[#8LT#2] THEN#8=#2; 限制下一次钻削深度不小于最小钻削深度

#9=#9-#8; 计算钻削深度剩余量

#11=#11-#8; 计算下一次钻削深度 Z 坐标值

END1; 循环 1 结束

G01 Z-#26; 直线切削到孔深

N10 G00 Z50; 顺序号 10 程序段: 快速上升到 Z=50 处

M99; 子程序结束

%

② G83 钻孔方式。

O6351; 子程序号

G00 Z#18; 快速下降到 R 平面

#10=#18+#26; 总钻削深度 (绝对值)

#8=#1; 给每次钻削深度赋初始值: 第一次钻削深度#1

#9=#10-#8; 第一次钻削后深度剩余量

#11=#18-#8; 第一次钻削深度 Z 坐标值

WHILE[#9GT0] DO1; 循环 1 设定条件: 钻削深度剩余量大于 0

G01 Z#11 F#7; 直线切削到钻削深度

IF[#11EQ-#26] GOTO10; 如果钻削深度等于孔深, 跳转到顺序号 10 程序段

G00 Z#18; 返回到 R 平面

```
Z[#3+#11]; 刀具快速下降到距上次切削底面一个回退量#3 处  
#8=#8-#6; 下一次钻削深度计算  
IF[#8LT#2] THEN#8=#2; 限制下一次钻削深度不小于最小钻削深度  
#9=#9-#8; 计算钻削深度剩余量  
#11=#11-#8; 计算下一次钻削深度 Z 坐标值  
END1; 循环 1 结束  
G01 Z-#26; 直线切削到孔深  
N10 G00 Z50; 顺序号 10 程序段: 快速上升到 Z=50 处  
M99; 子程序结束  
%
```

4) 主程序如下:

```
%  
O351;  
G54 G90 G00 G43 H1 Z50;  
X0 Y0;  
M03 S1000;  
G65 P6351 A15 B8 C1 K2 R5 D300 Z30; 调用宏程序 O6351, 给  
相关变量赋初始值  
  
G00 G49 Z100;  
X0 Y150;  
M05;  
M30;  
%
```

2. 等比数列深孔加工

如图 6-52 所示, 钻孔采用多次钻削, 第一孔钻深#1, 所有钻深成等比数列(系数为#6)递减变化规律, 最小钻深#2, 参照深孔循环指令 G73 和 G83 加工形式编写宏程序。

1) 分析: 如图 6-52a 所示, 孔加工切削流程: 水平 XY 位置快速定位, 快速下刀至 R 平面, 从 R 平面开始第一次钻削, 深度为#1, 但 Z 轴坐标值不是-#1, 而是-#1+#18, 快速抬刀#3, 接着进行第二次钻削, 下深距离为#1*#6 (#6 为等比系数), Z 轴坐标值为-#1+#18-#1*#6, 再抬刀#3, 接着进行第三次钻削, 下深距离为#1*#6*#6, 后面依此类推。

2) 变量及说明:

#1=[A]; 第一次钻深

#2=[B]; 最小钻深

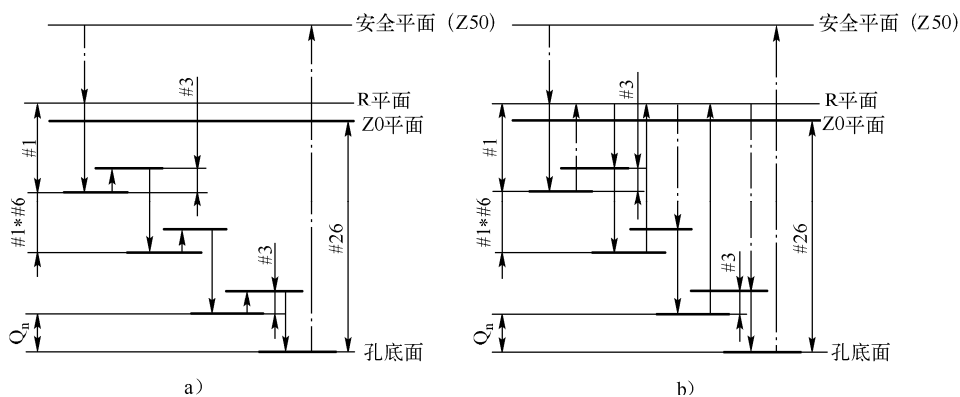


图 6-52 等比数列深孔加工

a) G73 b) G83

#3=[C]; 回退量

#6=[K]; 等比系数

#7=[D]; 进给量, 单位为 mm/min

#18=[R]; R 平面

#26=[Z]; 孔的深度

3) 宏程序如下:

① G73 钻孔方式。

%

O6352; 子程序号

G00 Z#18; 快速下降到 R 平面

#10=#18+#26; 总钻削深度 (绝对值)

#8=#1; 给每次钻削深度赋初始值: 第一次钻削深度#1

#9=#10-#8; 第一次钻削后深度剩余量

#11=#18-#8; 第一次钻削深度 Z 坐标值

WHILE[#9GT0] DO1; 循环 1 设定条件: 钻削深度剩余量大于 0

G01 Z#11 F#7; 直线切削到钻削深度

IF[#11EQ-#26] GOTO10; 如果钻削深度等于孔深, 跳转到顺序号 10 程序段

G00 Z[#3+#11]; 刀具退后一个回退量#3

#8=#8*#6; 下一次钻削深度计算

IF[#8LT#2] THEN#8=#2; 限制下一次钻削深度不小于最小钻削深度

#9=#9-#8; 计算钻削深度剩余量

#11=#11-#8; 计算下一次钻削深度 Z 坐标值

END1; 循环 1 结束

G01 Z-#26; 直线切削到孔深

N10 G00 Z50; 顺序号 10 程序段: 快速上升到 Z=50 处

M99; 子程序结束

%

② G83 钻孔方式。

%

O6352; 子程序号

G00 Z#18; 快速下降到 R 平面

#10=#18+#26; 总钻削深度 (绝对值)

#8=#1; 给每次钻削深度赋初始值: 第一次钻削深度#1

#9=#10-#8; 第一次钻削后深度剩余量

#11=#18-#8; 第一次钻削深度 Z 坐标值

WHILE[#9GT0] DO1; 循环 1 设定条件: 钻削深度剩余量大于 0

G01 Z#11 F#7; 直线切削到钻削深度

IF[#11EQ-#26] GOTO10; 如果钻削深度等于孔深, 跳转到顺序号 10 程序段

G00 Z#18; 快速上升到 R 平面

Z[#3+#11]; 刀具快速下降到距上次钻削底面一个回退量#3 处

#8=#8*#6; 下一次钻削深度计算

IF[#8LT#2] THEN#8=#2; 限制下一次钻削深度不小于最小钻削深度

#9=#9-#8; 计算钻削深度剩余量

#11=#11-#8; 计算下一次钻削深度 Z 坐标值

END1; 循环 1 结束

G01 Z-#26; 直线切削到孔深

N10 G00 Z50; 顺序号 10 程序段: 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

O352;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6352 A15 B8 C1 K0.9 R5 D300 Z30; 调用宏程序 O6352, 给
相关变量赋初始值

```

G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.4 矩形加工

在数控加工中，平面加工是最基本、最简单的加工方式。通常以加工后的上平面中心作为编程坐标系原点，所以首先需要加工好上平面，一般只要加工平整即可。

6.4.1 实例一矩形上平面加工

下面举一简单实例说明矩形上平面的加工方法，如图 6-53 所示。

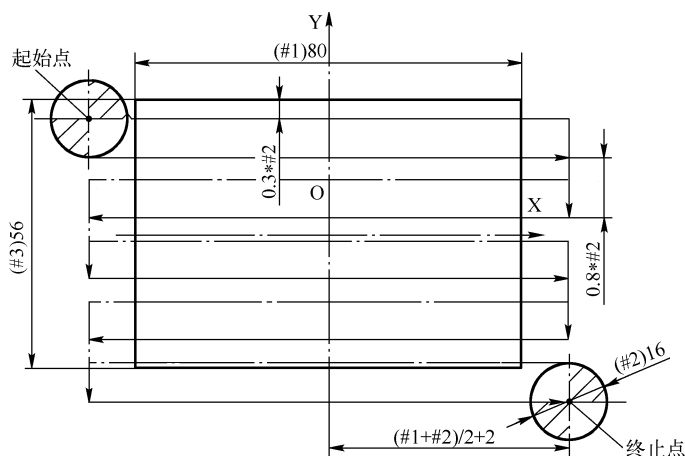


图 6-53 矩形上平面加工

(1) 深度方向一次成形

1) 分析：这是一个矩形面的加工，长、宽分别为#1、#3，深度方向只需要一次加工即可。

首先以工件上平面中心（已加工好的面）为坐标原点，X、Y、Z 三轴正向分别为向上、前、右。水平面需要采用左右往复切削，前后方向行间距按照铣刀直径的 0.8 来确定，即#8（行间距）=0.8*#2（#2 为铣刀直径），从工件左上角开始加工。

X 轴方向左右刀具中心极限位置为： $\pm[(\#1+\#2)/2+2]$ ，刀具中心左右切削

时过工件轮廓 2mm, Y 轴方向需要切削次数: $\#3/\#8$ 。

切削起始点 X: $-(\#1+\#2)/2-2$, Y: $\#3/2-0.3*\#2$, 利用上取整函数指令确定 Y 轴方向切削次数的编程: $\#11=\text{FIX}[\#3/\#8]$, Y 轴方向从第二轮切削时的坐标计算: 第一轮切削时的坐标递减一个行间距。

2) 变量及说明:

$\#1=[A]$; 矩形长度

$\#2=[B]$; 立铣刀直径

$\#3=[C]$; 矩形宽度

$\#7=[D]$; 进给量

3) 宏程序如下:

%

O6411; 子程序号

$\#8=0.8*\#2$; 设定刀间距 $=0.8\times$ 刀具直径

$\#5=[\#1+\#2]/2+2$; X 轴向加工极限点坐标值

$\#9=\#3/2-0.3*\#2$; 加工起始点 Y 轴向坐标值

$\#11=\text{FIX}[\#3/\#8]$; 计算水平铣削次数=工件宽度/刀间距, 上取整

G00 X- $\#5$ Y $\#9$; 水平快速定位到加工起始点

Z0; 下降到加工面高度

WHILE $[\#11\text{GE}0]$ DO1; 循环 1 设定条件: $\#11$ 不小于 0

G01 X $\#5$ F $\#7$; 水平向右铣削平面到右极限位置

$\#9=\#9-\#8$; 计算下一刀 Y 轴向坐标

$\#11=\#11-1$; 变量 $\#11$ 减去 1

IF $[\#11\text{LT}0]$ GOTO5; 如果 $\#11$ 小于 0 跳转到程序段顺序号 5 处

G00 Y $\#9$; 快速移动到下一加工起始点

G01 X- $\#5$; 水平向左铣平面到左极限位置

$\#9=\#9-\#8$; 计算下一刀 Y 轴向坐标

$\#11=\#11-1$; 变量 $\#11$ 减去 1

G00 Y $\#9$; 快速移动到下一加工起始点

END1; 循环 1 结束

N5 G00 Z50; 快速抬高刀具到 Z50

M99; 子程序结束

%

4) 主程序如下:

%

O411; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿
X0 Y0; 水平定位于 X0 Y0
M03 S1000; 主轴正转, 转速为 1000r/min
G65 P6411 A80 B16 C56 D300; 调用宏程序 O6411, 并赋相关变量初始值
G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100
X0 Y150; 刀具返回到水平加工起始位 X0 Y150
M05; 主轴停止
M30; 程序结束, 返回程序头
%

(2) 深度方向需多次切削

如果上平面在 Z 方向上需要多次深度分层加工, 则只需对上述 O6411 程序进行一定的修改即可, 具体见下面实例。

深度为 5mm, 其余条件同上例。

1) 分析: 深度设变量为 #4, 深度递增量每次 1.5mm, 设变量 #6。需要 #10=FIX[#4/#6] 次分层切削。每层切削方法和上例相同, 在宏程序中, 还要注意完成一层切削后, 在进行下一层加工前需要个别坐标值还原, 如: 加工起始点 Y 坐标值 #9、水平铣削次数值 #11。

2) 变量及说明:

#1=[A]; 矩形长度
#2=[B]; 立铣刀直径
#3=[C]; 矩形宽度
#4=[I]; 需要背吃刀量
#6=[K]; 深度递增量
#7=[D]; 进给量

3) 宏程序如下:

%
O6412; 子程序号
#8=0.8*#2; 设定刀间距=0.8×刀具直径
#5=[#1+#2]/2+2; X 轴向加工极限点坐标值
#9=#3/2-0.3*#2; 加工起始点 Y 轴向坐标值
#10=FIX[#4/#6]; 计算深度切削次数=深度/深度递增量, 上取整
#11=FIX[#3/#8]; 计算水平铣削次数=工件宽度/刀间距, 上取整
G00 X-#5 Y#9; 水平快速定位到加工起始点
Z5; 快速下降到 Z=5 处


```
#15=0; 给深度加工次数计数器赋初始值 0
WHILE[#15LE#10] DO2; 循环 2 设定条件: 计数器值不大于深度切削次数
#26=#6*[#15+1]; 计算每层背吃刀量值
IF[#26GT#4] THEN#26=#4; 如果算出的背吃刀量值大于深度#4, 则限定它
                        最大只能是#4
G01 Z-#26 F[0.3*#7]; 直线切削到深度#26
#11=FIX[#3/#8]; 计算水平铣削次数值还原
WHILE[#11GE0] DO1; 循环 1 设定条件: #11 不小于 0
G01 X#5 F#7; 水平向右铣削平面到右极限位置
#9=#9-#8; 计算下一刀 Y 轴向坐标
#11=#11-1; 变量#11 减去 1
IF[#11LT0] GOTO5; 如果#11 小于 0 跳转到程序段顺序号 5 处
G00 Y#9; 快速移动到下一加工起始点
G01 X-#5; 水平向左铣平面到左极限位置
#9=#9-#8; 计算下一刀 Y 轴向坐标
#11=#11-1; 变量#11 减去 1
G00 Y#9; 快速移动到下一加工起始点
END1; 循环 1 结束
N5 G00 Z[2-#26]; 顺序号 5 程序段: 快速抬刀 2mm
#9=#3/2-0.3*#2; 还原加工起始点 Y 轴向坐标值
X-#5 Y#9; 返回到每层加工起始点
#15=#15+1; 深度切削次数计数器值累加 1
END2; 循环 2 结束
G00 Z50; 快速抬高刀具到 Z50
M99; 子程序结束
%
4) 主程序如下:
%
O412; 主程序号
G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿
X0 Y0; 水平定位于 X0 Y0
M03 S1000; 主轴正转, 转速为 1000r/min
G65 P6412 A80 B16 C56 I5 K1.5 D300; 调用宏程序 O6412, 并赋
                                相关变量初始值
G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100
```

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

6.4.2 实例二矩形槽加工

1. 直底矩形槽粗加工

1) 分析: 与上例矩形上平面相比, 矩形槽属于封闭区域平面加工, 在编程方面的要求和约束要严。如图 6-54 所示, 矩形槽长度为#1, 宽度为#3, 深度为#5, 内圆角半径为#4。现在用直径为#2 立铣刀进行铣削粗加工, 侧壁和槽底间边精加工余量都是#9, 分层铣削, 每次深度为#6, 共需#10=FIX[[(#6-#9)/#6]次完成深度加工。

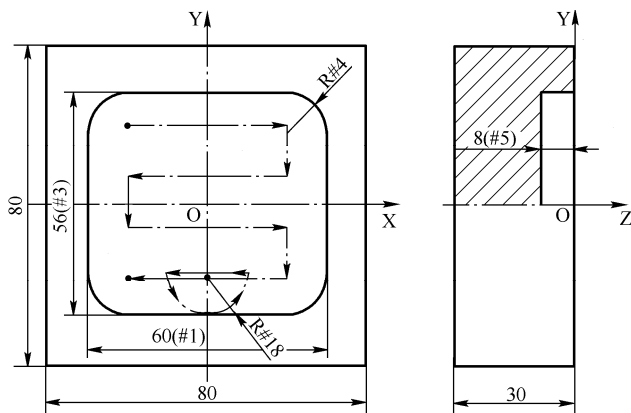


图 6-54 直底矩形槽加工

加工方式: 采用分层铣削, 先中间后四周侧壁, 完成一层再进行下一层的加工, 直至加工到要求的深度。每层下刀点都是左上角, 坐标如下: $(-[\#1/2-\#9-\#4], [\#3/2-\#9-\#4])$ 。由于是轴向向下刀, 可以先用比铣削用刀具直径稍小的钻头预钻孔, 或用切削刃过中心的刀具, 只是 Z 向下刀速度一定要足够慢。

中间平面采用水平左右往复铣削, 刀具中心左右极限为 $\pm[\#1/2-\#9-\#4]$, Y 轴方向需要切削次数为 $\text{FIX}[(\#3-\#9*2)/\#8]$, Y 轴方向从第二轮切削时的坐标计算: 第一轮切削时的坐标递减一个行间距, 直至 $-\#3/2-\#9-\#4$ 。槽四周侧壁运用半径补偿, 以槽侧轮廓尺寸进行编程, 从槽中间后侧开始, 逆时针走刀路线, 完成槽四周的粗加工。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 圆角半径

#5=[J]; 槽深度

#6=[K]; 深度递增量

#7=[D]; 进给量

#9=[F]; 精加工单边余量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6421; 子程序号

#8=0.8*#2; 设定刀间距=0.8×刀具直径

#14=#1/2-#9-#4; 计算 X 轴向极限坐标值

#15=#3/2-#9-#4; 计算 Y 轴向极限坐标值

#10=FIX[#6-#9]/#6; 计算深度切削次数

#11=FIX[2*#15/#8]; 计算水平切削往复次数

G00 X0 Y0; 快速定位到 X0 Y0

Z5; 快速下降到 Z=5 处

WHILE[#10GE0] DO2; 循环 2 设定条件: 深度切削次数#10 不小于 0

#26=#6-#6*#10; 计算每层背吃刀量

G00 X[2-#14] Y[#16-2]; 快速定位到加工准备位置

G01 X-#14 Y#15 F#7; 直线切削到加工起始点

Z-#26; 直线切削下降到背吃刀量

#25=#15; 给加工起始点 Y 轴坐标赋初始值

G01 X#14; 直线切削到右极限位置

#16=1; 循环 1 往复次数计数器#16 赋初始值 1

WHILE[#16LT#11] DO1; 循环 1 设定条件: 变量#16 小于计算的往复次数

#25=#16-#8*#16; 计算每轮水平切削 Y 轴坐标

IF[#25LT-#15] THEN#25=-#15; 限定 Y 轴坐标值最小值等于 Y 轴负极限尺寸

G01 Y#25; 直线切削到往复切削下轮起始点

G01 X-#14; 直线切削到左极限位置

#16=#16+1; 变量#16 累加 1

IF[#16GE#11] GOTO5; 如果变量#16 值大于等于计算的往复次数跳转到顺
序号 5 程序段

#25=#16-#8*#16; 计算每轮水平切削 Y 轴坐标

IF[#25LT-#15] THEN#25=-#15; 限定 Y 轴坐标值最小值等于 Y 轴负极限尺寸
G01 Y#25; 直线切削到往复切削下轮起始点
G01 X#14; 直线切削到右极限位置
#16=#16+1; 变量#16 累加 1
END1; 循环 1 结束
N5 G00 Z[2-#26]; 顺序号 5 程序段: 快速上升 2mm
X0 Y[#18-1-#15]; 快速定位到槽侧壁轮廓加工起始位置
G41 D1 X-#18; 半径补偿
G03 X0 Y-[#3/2] Z-#26 R#18; 圆弧切入
G01 X[#1/2], R4; 槽侧壁轮廓加工, 倒右下角圆角
Y[#3/2], R4; 槽侧壁轮廓加工, 倒右上角圆角
X-[#1/2], R4; 槽侧壁轮廓加工, 倒左上角圆角
Y-[#3/2], R4; 槽侧壁轮廓加工, 倒左下角圆角
X0; 完成槽侧壁轮廓加工
G03 X#18 Y[#18-1-#15] R#18; 圆弧切出
G00 Z[2-#26]; 快速抬刀 2mm
G40 X0 Y0; 取消半径补偿
#10=#10-1; 深度计数器计数累减 1
END2; 循环 2 结束
N5 G00 Z50; 快速抬高刀具到 Z50
M99; 子程序结束
%
4) 主程序如下:
%
O421; 主程序号
G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿
X0 Y0; 水平定位于 X0 Y0
M03 S400; 主轴正转, 转速为 400r/min
G65 P6421 A60 B16 C56 I10 J8 K1.5 D300 R10; 调用宏程序 O6421,
并赋相关变量初始值
G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100
X0 Y150; 刀具返回到水平加工起始位 X0 Y150
M05; 主轴停止
M30; 程序结束, 返回程序头
%

2. 直底矩形槽精加工

1) 分析: 如图 6-54 所示, 首先进行槽底的精加工, 然后进行槽四周侧壁的精加工。左右往复铣削, X、Y 轴方向最大极限尺寸分别为: $[(\#1-\#2)/2-\#4]$ 、 $[(\#3-\#2)/2-\#4]$, 其他相关尺寸以及加工路线等参照粗加工; 再从槽中间后侧圆弧切入, 运用半径补偿, 以槽侧轮廓尺寸进行编程, 完成槽侧的四周精加工。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 圆角半径

#5=[J]; 槽深度

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6422; 子程序号

#20=0.8*#2; 计算行间距

#11=FIX $[(\#3-2*\#4)/\#20]$; 计算槽底面水平往复切削次数

G00 X $[(\#1-\#2)/2-\#4]$ Y $[(\#3-\#2)/2-\#4]$; 快速定位到左上角加工起始位置
Z $[2-\#5]$; 快速下降到离底面 2mm 处

G01 Z $\#5$ F $\#7$; 直线切削至深度

#17=0; 给往复次数计数器变量#17 赋初始值 0

WHILE[#17LE#11] DO1; 循环 1 设定条件: 变量#17 不大于往复次数

#25= $[(\#3-\#2)/2-\#4-\#20*\#17]$; 计算每轮切削 Y 轴坐标值

#35= $[(\#3-\#2)/2-\#4]$; 计算 Y 轴坐标最小极限尺寸

IF[#25LT#35] THEN#25=#35; 限定 Y 轴坐标不小于最小极限尺寸

G01 Y#25; 直线切削到往复切削起始点

X $[(\#1-\#2)/2-\#4]$; 水平向右切削至右极限位置

#17=#17+1; 变量#17 累加 1

IF[#17GT#11] GOTO5; 如果变量#17 值大于往复次数跳转到序号 5 程序段

#25= $[(\#3-\#2)/2-\#4-\#20*\#17]$; 计算每轮切削 Y 轴坐标值

IF[#25LT#35] THEN#25=#35; 限定 Y 轴坐标不小于最小极限尺寸

G01 Y#25; 直线切削到往复切削起始点

X $[(\#1-\#2)/2-\#4]$; 水平向左切削至左极限位置

#17=#17+1; 变量#17 累加 1

END1; 循环 1 结束

N5 G00 Z[2-#5]; 顺序号 5 程序段: 快速抬刀 2mm

X0 Y[#18-#3/2]; 快速定位到槽轮廓加工准备位置

G41 D1 X-#18; 半径补偿

G03 X0 Y-[#3/2] Z-#5 R#18; 螺旋切入

G01 X[#1/2], R#4; 槽侧壁精加工, 倒右下角圆角

Y[#3/2], R#4; 槽侧壁精加工, 倒右上角圆角

X-[#1/2], R#4; 槽侧壁精加工, 倒左上角圆角

Y-[#3/2], R#4; 槽侧壁精加工, 倒左下角圆角

X0; 完成槽侧壁精加工

G03 X#18 Y[#18-#3/2] Z[2-#5] R#18; 螺旋切出

G00 Z50; 快速抬高刀具到 Z50

G40 X0 Y0; 取消半径补偿

M99; 子程序结束

%

4) 主程序如下:

%

O422; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿

X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6422 A60 B16 C56 I10 J8 D1000 R10; 调用宏程序 O6422, 并
赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

3. 圆弧底矩形槽粗加工

1) 分析: 如图 6-55 所示, 矩形槽长度为#1, 宽度为#3, 深度为#5, 内圆角半径为#4, 槽底四周有半径为#8 圆角。现在用直径为#2 立铣刀进行铣削粗加工, 侧壁和槽底间边都保留精加工余量#9, 分层铣削, 每次深度为#6, 共需#10=FIX[#6-#9]/#6次完成深度加工。与上例直底矩形槽相比, 槽底多了一个圆角, 在铣削到槽底时, 如果用圆鼻刀, 刀尖圆角半径正好等于槽底圆角半径#8, 编程就方便了, 所以通常尽可能地选择这样的刀具。当用立铣刀加工时, 四角尺寸计算

必须考虑细致，中间平面铣削到槽底时，留有圆角余量，槽四侧壁铣削时，不能铣到槽底，而是#6-#8。

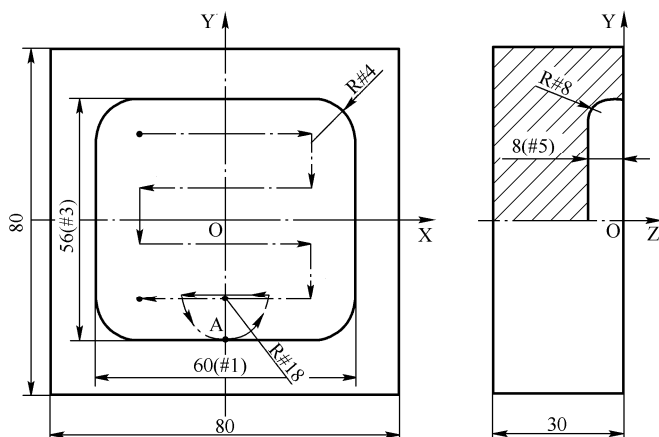


图 6-55 圆弧底矩形槽粗加工

加工方式：采用分层铣削，先中间后四周侧壁，完成一层再进行下一层的加工，直至加工到要求的深度，最后再去除槽底四周圆角余量。每层下刀点都是左上角，坐标为 $(-[\#1-\#2]/2-\#8)$, $[\#3-\#2]/2-\#8$)。由于是轴向下刀，可以先用比铣削刀具直径稍小的钻头预钻孔，或用切削刃过中心的刀具，只是 Z 向下刀速度一定要足够慢。

中间平面采用水平左右往复铣削，刀具中心左右极限为 $\pm [[\#1-\#2]/2-\#8]$ ，Y 轴方向需要切削次数为 $\text{FIX}[[\#3-\#8*2]/\#20]$ ，Y 轴方向从第二轮切削时的坐标计算：第一轮切削时的坐标递减一个行间距，直至 $-[[\#3-\#2]/2-\#8]$ 。槽四周侧壁运用半径补偿，以槽侧轮廓尺寸进行编程，从槽中间后侧开始，逆时针走刀路线，深度控制在 #6-#8，完成槽四侧的粗加工。槽底四周圆角余量，采用从槽底向上、向外四周逐层铣削去除，以切削点与垂直面夹角 #17 为变量，#6 (#6=10°) 为递增量，随着角度 #17 增大，切削点沿四周圆角向上、向外扩散加工。

2) 变量及说明：

- #1=[A]；矩形长度
- #2=[B]；立铣刀直径
- #3=[C]；矩形宽度
- #4=[I]；圆角半径
- #5=[J]；槽深度
- #6=[K]；深度递增量
- #7=[D]；进给量

#8=[E]; 槽底圆角半径

#9=[F]; 精加工单边余量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6423; 子程序号

#20=0.8*#2; 设定刀间距=0.8×刀具直径

#14=[#1-#2]/2-#8; 计算 X 轴向最大极限坐标值

#15=[#3-#2]/2-#8; 计算 Y 轴向最大极限坐标值

#10=FIX[[#6-#9]/#6]; 计算深度切削次数

#11=FIX[2*#15/#20]; 计算水平切削往复次数

G00 X0 Y0; 快速定位到 X0 Y0

Z5; 快速下降到 Z=5 处

WHILE[#10GE0] DO2; 循环 2 设定条件: 深度切削次数#10 不小于 0

#26=#6-#6*#10; 计算每层背吃刀量

G00 X-#14 Y#15; 直线切削到加工起始点

G01 Z-#26 F#7; 直线切削下降到背吃刀量

#25=#15; 给加工起始点 Y 轴坐标赋初始值

G01 X#14; 直线切削到右极限位置

#16=1; 循环 1 往复次数计数器#16 赋初始值 1

WHILE[#16LT#11] DO1; 循环 1 设定条件: 变量#16 小于计算的往复次数

N5 #25=#16-#20*#16; 计算每轮水平切削 Y 轴坐标

IF[#25LT-#15] THEN#25=-#15; 限定 Y 轴坐标值最小值等于 Y 轴负极限尺寸

G01 Y#25; 直线切削到往复切削下轮起始点

G01 X-#14; 直线切削到左极限位置

#16=#16+1; 变量#16 累加 1

IF[#16GE#11] GOTO5; 如果变量#16 值大于等于计算的往复次数跳转到顺序号 15 程序段

#25=#16-#20*#16; 计算每轮水平切削 Y 轴坐标

IF[#25LT-#15] THEN#25=-#15; 限定 Y 轴坐标值最小值等于 Y 轴负极限尺寸

G01 Y#25; 直线切削到往复切削下轮起始点

G01 X#14; 直线切削到右极限位置

#16=#16+1; 变量#16 累加 1

END1; 循环 1 结束

N15 G00 Z[2-#26]; 顺序号 15 程序段: 快速上升 2mm

IF[#26GT#6-#8] GOTO5; 如果背吃刀量#26 大于限定深度, 则跳转到顺序号 5 程序段

X0 Y[#18-#3/2+#9]; 快速定位到槽侧壁轮廓加工起始位置

G41 D1 X-#18; 半径补偿

G03 X0 Y-[#3/2-#9] Z-#26 R#18; 螺旋切入

G01 X[#1/2-#9], R#4; 槽侧壁轮廓加工, 倒右下角圆角

Y[#3/2-#9], R#4; 槽侧壁轮廓加工, 倒右上角圆角

X-[#1/2-#9], R#4; 槽侧壁轮廓加工, 倒左上角圆角

Y-[#3/2-#9], R#4; 槽侧壁轮廓加工, 倒左下角圆角

X0; 完成槽侧壁轮廓加工

G03 X#18 Y[#18-#3/2+#9] Z[2-#26] R#18; 螺旋切出

G00 Z[2-#26]; 快速抬刀 2mm

G40 X0; 取消半径补偿

#10=#10-1; 深度计数器计数累减 1

END2; 循环 2 结束

#6=10; 给角度变量增量赋值 10°

#17=#6; 给角度变量赋初始值#6

WHILE[#17LE90] DO1; 循环 1 设定条件: 角度变量不大于 90°

#26=#6-#8*[1-COS*[#17]]-#9; 计算相对于角度变量切削点 Z 轴坐标

#24=#1/2-#8*[1-SIN*[#17]]-#9; 计算相对于角度变量切削点 X 轴坐标

#25=#3/2-#8*[1-SIN*[#17]]-#9; 计算相对于角度变量切削点 Y 轴坐标

G41 D1 G00 X-#18 Y-[#26-#18]; 半径补偿

G03 X0 Y-#25 Z-#26 R#18; 螺旋切入

G01 X#24, R#4; 槽侧壁轮廓加工, 倒右下角圆角

Y#25, R#4; 槽侧壁轮廓加工, 倒右下角圆角

X-#24, R#4; 槽侧壁轮廓加工, 倒右下角圆角

Y-#25, R#4; 槽侧壁轮廓加工, 倒右下角圆角

X0; 完成槽侧壁轮廓加工

G03 X#18 Y-[#26-#18] Z[2-#26] R#18; 螺旋切出

G40 G00 X0; 取消半径补偿

#17=#17+#6; 角度变量累加递增量#6

END1; 循环 1 结束

G00 Z50; 快速抬高刀具到 Z50

M99; 子程序结束

%

4) 主程序如下:

%

O423; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿
X0 Y0; 水平定位于 X0 Y0

M03 S400; 主轴正转, 转速为 400r/min

G65 P6423 A60 B16 C56 I10 J8 K1.5 D300 E3 F0.3 R10; 调用宏程序 O6423,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停

M30; 程序结束, 返回程序头

%

4. 圆弧底矩形槽精加工

(1) 立铣刀

1) 分析: 如图 6-56 所示, 首先进行槽底的精加工, 然后进行槽底四周圆角侧壁的精加工。槽底精加工采用左右往复铣削, X、Y 轴最大极限尺寸分别为: ($\#1/2-\#4$)、($\#3/2-\#4$), 再从槽中间后侧圆弧切入, 运用半径补偿, 从槽底圆角起始处进行精加工, 逐层向上精加工, 以切削点与垂直面倾斜角为变量, 该角度变量从 $0^\circ \sim 90^\circ$ 。这里刀具的参考点是刀具中心顶点, 坐标计算要注意, 如图 6-56a

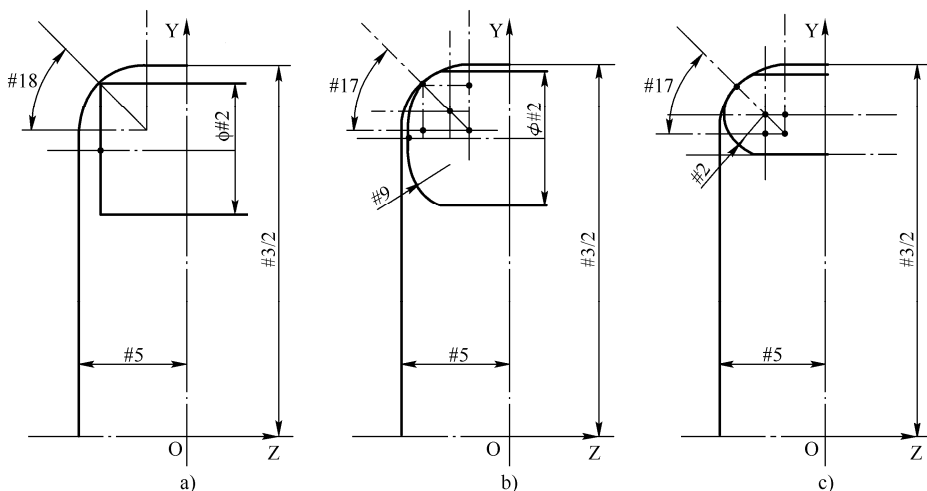


图 6-56 圆弧底矩形槽精加工局部放大图

a) 立铣刀加工 b) 圆鼻刀加工 c) 球头铣刀加工

所示,运用数学几何知识,计算出刀具参考点相应坐标轴的坐标值,进而完成槽底圆周和槽侧的四周精加工。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 圆角半径

#5=[J]; 槽深度

#6=[K]; 角度递增量

#7=[D]; 进给量

#8=[E]; 槽底圆角半径

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6424; 子程序号

#20=0.8*#2; 计算行间距

#11=FIX[(#3-2*#4)/#20]; 计算槽底面水平往复切削次数

G00 X-[(#1-#2)/2-#4] Y[(#3-#2)/2-#4]; 快速定位到左上角加工起始位置

Z[2-#5]; 快速下降到离底面 2mm 处

G01 Z-#5 F#7; 直线切削至深度

#17=0; 给往复次数计数器变量#17 赋初始值 0

WHILE[#17LE#11] DO1; 循环 1 设定条件: 变量#17 不大于往复次数

#25=[#3-#2)/2-#4-#8*#17; 计算每轮切削 Y 轴坐标值

#35=-[(#3-#2)/2-#4]; 计算 Y 轴坐标最小极限尺寸

IF[#25LT#35] THEN#25=#35; 限定 Y 轴坐标不小于最小极限尺寸

G01 Y#25; 直线切削到往复切削起始点

X[(#1-#2)/2-#4]; 水平向右切削至右极限位置

#17=#17+1; 变量#17 累加 1

IF[#17GT#11] GOTO5; 如果变量#17 值大于往复次数, 跳转到顺序号 5 程序段

#25=[#3-#2)/2-#4-#8*#17; 计算每轮切削 Y 轴坐标值

IF[#25LT#35] THEN#25=#35; 限定 Y 轴坐标不小于最小极限尺寸

G01 Y#25; 直线切削到往复切削起始点

X-[(#1-#2)/2-#4]; 水平向左切削至左极限位置

#17=#17+1; 变量#17 累加 1

END1; 循环 1 结束

N5 G00 Z[2-#5]; 顺序号 5 程序段: 快速抬刀 2mm
#17=0; 角度变量#17 赋初始值 0
WHILE[#17LE90] DO1; 循环 1 设定条件: 角度变量不大于 90°
#26=#6-#8*[1-COS*[#17]]; 计算对应于角度变量切削点的 Z 轴坐标
#24=#1/2-#8*[1-SIN*[#17]]; 计算对应于角度变量切削点的 X 轴坐标
#25=#3/2-#8*[1-SIN*[#17]]; 计算对应于角度变量切削点的 Y 轴坐标
G00 X0 Y[#18+#8-#3/2]; 快速定位到槽轮廓加工准备位置
G41 D1 X-#18 Y[#18-#25]; 半径补偿
G03 X0 Y-#25 Z-#26 R#18; 螺旋切入
G01 X#24 R#4; 槽底圆角精加工
Y#25 R#4; 槽底圆角精加工
X-#24 R#4; 槽底圆角精加工
Y-#25 R#4; 槽底圆角精加工
X0; 槽底圆角精加工
G03 X#18 Y[#18-#25] Z[2-#26] R#18; 螺旋切出
G40 G00 X0; 取消半径补偿
#17=#17+#6; 角度变量累加一个递增量#6
END1; 循环 1 结束
G00 Z50; 快速抬高刀具到 Z50
M99; 子程序结束

%

4) 主程序如下:

%

O424; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿

X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6424 A60 B16 C56 I10 J8 K1.5 D300 E3 F0.3 R10; 调用宏程序 O6424,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停

M30; 程序结束, 返回程序头

%

(2) 圆鼻刀

1) 分析: 如图 6-56 所示, 首先进行槽底的精加工, 然后进行槽底四周圆角侧壁的精加工。槽底精加工采用左右往复铣削, X、Y 轴最大极限尺寸分别为: ($\#1/2-\#4$)、($\#3/2-\#4$), 再从槽中间后侧圆弧切入, 运用半径补偿, 从槽底圆角起始处进行精加工, 逐层向上精加工, 以切削点与垂直面倾斜角为变量, 该角度变量从 $0^\circ \sim 90^\circ$ 。这里圆鼻刀的参考点是刀具中心顶点, 坐标计算比较烦琐, 如图 6-56b 所示, 运用数学几何知识, 计算出刀具参考点相应坐标轴的坐标值, 进而完成槽底圆周和槽侧的四周精加工。

2) 变量及说明:

$\#1=[A]$; 矩形长度

$\#2=[B]$; 立铣刀直径

$\#3=[C]$; 矩形宽度

$\#4=[I]$; 圆角半径

$\#5=[J]$; 槽深度

$\#6=[K]$; 角度递增量

$\#7=[D]$; 进给量

$\#8=[E]$; 槽底圆角半径

$\#9=[F]$; 刀尖角半径 r

$\#18=[R]$; $1/4$ 圆弧切入切出半径

3) 宏程序如下:

%

O6425; 子程序号

$\#20=0.8*\#2$; 计算行间距

$\#11=FIX[(\#3-2*\#4)/\#20]$; 计算槽底面水平往复切削次数

G00 X $[(\#1-\#2)/2-\#4]$ Y $[(\#3-\#2)/2-\#4]$; 快速定位到左上角加工起始位置

Z $[2-\#5]$; 快速下降到离底面 2mm 处

G01 Z $-\#5$ F $\#7$; 直线切削至深度

$\#17=0$; 给往复次数计数器变量 $\#17$ 赋初始值 0

WHILE $[\#17LE\#11]$ DO1; 循环 1 设定条件: 变量 $\#17$ 不大于往复次数

$\#25=(\#3-\#2)/2-\#4-\#8*\#17$; 计算每轮切削 Y 轴坐标值

$\#35=-[(\#3-\#2)/2-\#4]$; 计算 Y 轴坐标最小极限尺寸

IF $[\#25LT\#35]$ THEN $\#25=\#35$; 限定 Y 轴坐标不小于最小极限尺寸

G01 Y $\#25$; 直线切削到往复切削起始点

X $[(\#1-\#2)/2-\#4]$; 水平向右切削至右极限位置

$\#17=\#17+1$; 变量 $\#17$ 累加 1

IF $[\#17GT\#11]$ GOTO5; 如果变量 $\#17$ 值大于往复次数, 跳转到顺序号 5 程序段

#25=[#3-#2]/2-#4-#8*#17; 计算每轮切削 Y 轴坐标值
IF[#25LT#35] THEN#25=#35; 限定 Y 轴坐标不小于最小极限尺寸
G01 Y#25; 直线切削到往复切削起始点
X-[(#1-#2)/2-#4]; 水平向左切削至左极限位置
#17=#17+1; 变量#17 累加 1
END1; 循环 1 结束
N5 G00 Z[2-#5]; 序号 5 程序段: 快速抬刀 2mm
#17=0; 角度变量#17 赋初始值 0
WHILE[#17LE90] DO1; 循环 1 设定条件: 角度变量不大于 90°
#26=#5+[#9-#8]*[1-COS*[#17]]+#9; 计算对应于角度变量切削点的 Z 轴坐标
#24=#1/2+[#9-#8]*[1-SIN*[#17]]; 计算对应于角度变量切削点的 X 轴坐标
#25=#3/2+[#9-#8]*[1-SIN*[#17]]; 计算对应于角度变量切削点的 Y 轴坐标
G00 X0 Y[#18+#8-#3/2]; 快速定位到槽轮廓加工准备位置
G41 D1 X-#18 Y[#18-#25]; 半径补偿
G03 X0; Y-#25 Z-#26 R#18; 螺旋切入
G01 X#24, R#4; 槽底圆角精加工
Y#25, R#4; 槽底圆角精加工
X-#24, R#4; 槽底圆角精加工
Y-#25, R#4; 槽底圆角精加工
X0; 槽底圆角精加工
G03 X#18 Y[#18-#25] Z[2-#26] R#18; 螺旋切出
G40 G00 X0; 取消半径补偿
#17=#17+#6; 角度变量累加一个递增量#6
END1; 循环 1 结束
G00 Z50; 快速抬高刀具到 Z50
M99; 子程序结束
%
4) 主程序如下:
%
O425; 主程序号
G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿
X0 Y0; 水平定位于 X0 Y0
M03 S1000; 主轴正转, 转速为 1000r/min
G65 P6425 A60 B16 C56 I10 J8 K5 D300 E5 F3 R10; 调用宏程序 O6425, 并
赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

(3) 球头铣刀

只用于精加工槽底四周圆角及侧壁。

1) 分析: 如图 6-55 和图 6-56c 所示, 从槽中间后侧圆弧切入, 从槽底圆角起始处进行精加工, 逐层向上精加工, 以切削点与垂直面倾斜角为变量, 该角度变量从 $0^{\circ} \sim 90^{\circ}$ 。这里球头铣刀的参考点是刀具球心, 所以不用半径补偿, 坐标计算比较烦琐, 运用数学几何知识, 计算出刀具参考点相应坐标轴的坐标值, 进而完成槽底圆周的四精加工。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 圆角半径

#5=[J]; 槽深度

#6=[K]; 角度递增量

#7=[D]; 进给量

#8=[E]; 槽底圆角半径

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6426; 子程序号

G00 X0 Y0; 快速定位到槽中心位置 (0, 0)

Z-[2-#5]; 快速下降到离槽底 2mm 处

#17=0; 给角度变量赋初始值 0

WHILE[#17LE90] DO1; 循环 1 设定条件: 角度变量不大于 90°

#26=#6-#8+[#8-#2]*COS[#17]; 计算对应于角度变量任意值时球头铣刀球心
Z 轴坐标

#24=#1/2-#8*SIN[#17]; 计算对应于角度变量任意值时球头铣刀球心 X 轴坐标

#25=#3/2-#8*SIN[#17]; 计算对应于角度变量任意值时球头铣刀球心 Y 轴坐标

G00 X-#18 Y[#18-#25]; 快速定位到螺旋切入开始位置

G03 X0 Y-#25 Z-#26 R#18 F#7; 螺旋切入

```
G01 X#24, R#4; 槽底圆角精加工
Y#25, R#4; 槽底圆角精加工
X-#24, R#4; 槽底圆角精加工
Y-#25, R#4; 槽底圆角精加工
X0; 槽底圆角精加工
G03 X#18 Y[#18-#25] Z[2-#26] R#18; 螺旋切出
G0 X0; 返回加工起始点
#17=#17+#6; 角度变量累加一个递增量#6
END1; 循环1结束
G00 Z50; 快速上升到 Z=50
M99; 子程序结束
%
```

4) 主程序如下:

```
%
O426;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6426 A60 B3 C56 I10 J8 K10 D1000 E5 R10; 调用宏程序O6426,
                                                赋相关变量初始值
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

6.4.3 实例三矩形倒圆加工

1. 直角外矩形倒圆加工

(1) 立铣刀

1) 分析: 如图 6-57 所示, 矩形体长、宽分别为#1、#3, 上平面四周倒圆半径为#4, 现在用直径为#2 的立铣刀进行倒圆加工。

首先确定坐标系, 以工件上平面中心为坐标原点, 向右、向前、向上分别为 X、Y、Z 轴的正向。加工起始点从工件正前方, 中间点圆角与侧面切点(0, #3/2, -#4) 开始圆弧切入, 切入圆弧半径为#18; 顺时针走刀路线, 完成一层切削后, 圆弧切出。为了编程坐标计算方便, 运用半径补偿功能, 直接用轮廓相关尺寸进

行编程，一层一层向上分层切削，加工路线不变，在这里以切削点与倒圆中心连线的倾斜角#8 为变量，范围从 $0^\circ \sim 90^\circ$ 。X、Y 坐标尺寸按照#1 (#3) -#4*[1-COS[#8]] 规律缩小，深度尺寸变化规律是：#4*[1-SIN[#8]]，绝对值递减。

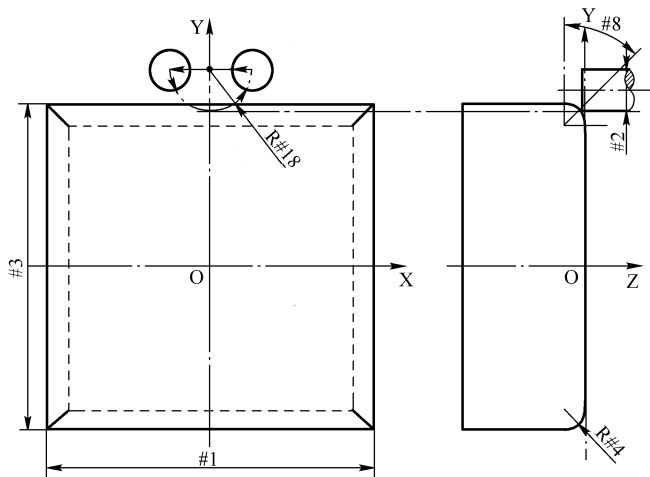


图 6-57 直角外矩形倒圆加工 (1)

运用半径补偿就可利用轮廓尺寸进行编程，切削是从下往上，当半径补偿值固定为#2/2 时，X、Y 轴坐标就不能固定，而分别是 $\pm[\#1/2-\#4*[1-\text{COS}[\#8]]]$ 、 $\pm[\#3/2-\#4*[1-\text{COS}[\#8]]]$ ，并且圆弧切入切出点以及开始、结束点坐标都是变化的，即切入切出点 (0, #3/2-#4*[1-COS[#8]])，圆弧切入开始点 (-#18, #3/2-#4*[1-COS[#8]]+#18)、圆弧切出结束点 (#18, #3/2-#4*[1-COS[#8]]+#18)。如果用 G10 P1 L12 R#10 指令，就可以即时给半径补偿值指定变化后的数值，半径补偿值 (D1) 的变化规律： $\#10=\#2/2-\#4*[1-\text{COS}[\#8]]$ ，这样编程坐标计算方便多了。

2) 变量及说明：

#1=[A]；矩形长度

#2=[B]；立铣刀直径

#3=[C]；矩形宽度

#4=[I]；倒圆 R 值

#6=[K]；递增量

#7=[D]；进给量

#18=[R]；1/4 圆弧切入切出半径

3) 宏程序如下：

① 用 G10 P1 L12 R#10 指令格式编写的宏程序。

%

O6431; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°

#26=#4-#4*SIN[#8]; 计算背吃刀量

#10=#2/2-#4+#4*COS[#8]; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器

G00 Z-#26; 快速下降到背吃刀量

G41 D1 X-#18; 半径补偿

G03 X0 Y[#3/2] R#18 F#7; 圆弧切入

G01 X[#1/2]; 轮廓加工

Y-[#3/2]; 轮廓加工

X-[#1/2]; 轮廓加工

Y[#3/2]; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#3/2+#18] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#8=#8+#6; 角度变量累加#6

END1; 循环 1 结束

M99; 子程序结束

%

② 用固定半径补偿值指令格式编写的宏程序。

%

O4312; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°

#26=#4-#4*SIN[#8]; 计算背吃刀量

#20=#4*[1-COS[#8]]; 修正值

G00 Z-#26; 快速下降到背吃刀量

G41 D1 X-#18 Y[#3/2-#20+#18]; 半径补偿

G03 X0 Y[#3/2-#20] R#18 F#7; 圆弧切入

G01 X[#1/2-#20]; 轮廓加工

Y-[#3/2-#20]; 轮廓加工

X-[#1/2-#20]; 轮廓加工

Y[#3/2-#20]; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#3/2-#20] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#8=#8+#6; 角度变量累加#6

END1; 循环 1 结束

M99; 子程序结束

%

4) 主程序如下:

%

O431; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿

X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6431 A80 B16 C80 I5 K10 D1000 R10; 调用宏程序 O6431,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

(2) 球头铣刀

1) 分析: 如图 6-58 所示, 矩形体长、宽分别为#1、#3, 上平面四周倒圆半径为#4, 现在用半径为#2 的球头铣刀进行倒圆加工。编程思路参照图 6-57 实例, 要注意: 参考点是球心, 倒圆时深度和水平轮廓的尺寸变化所涉及的半径是[#2+#4], 深度变化规律: $\#26=[\#2+\#4]*\sin[\#8]-\#4$, 半径补偿值 $\#10=[\#2+\#4]*\cos[\#8]-\#4$ 。这里用的是 G10 指令格式, 固定半径补偿值编程读者可自行尝试一下。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 球头铣刀半径

#3=[C]; 矩形宽度

#4=[I]; 倒圆 R 值

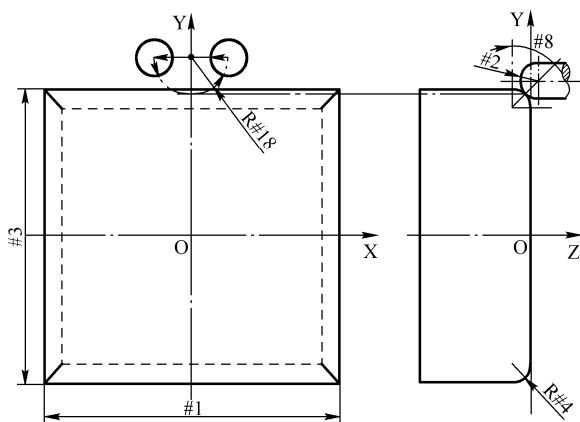


图 6-58 直角外矩形倒圆加工 (2)

#6=[K]; 递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6432; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°

#26=[#2+#4]*SIN[#8]-#4; 计算背吃刀量

#10=[#2+#4]*COS[#8]-#4; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器

G00 Z#26; 快速下降到背吃刀量

G41 D1 X-#18; 半径补偿

G03 X0 Y[#3/2] R#18 F#7; 圆弧切入

G01 X[#1/2]; 轮廓加工

Y-[#3/2]; 轮廓加工

X-[#1/2]; 轮廓加工

Y[#3/2]; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#3/2+#18] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#8=#8+#6; 角度变量累加#6

END1; 循环 1 结束

M99; 子程序结束

%

4) 主程序如下:

%

O432; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿

X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6432 A80 B8 C80 I5 K10 D1000 R10; 调用宏程序 O6432,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

2. 圆角外矩形倒圆加工

(1) 立铣刀

1) 分析: 如图 6-59 所示, 矩形体长、宽分别为#1、#3, 矩形体四周倒圆半径为#5, 上平面四周倒圆半径为#4, 现在用直径为#2 的立铣刀进行倒圆加工。

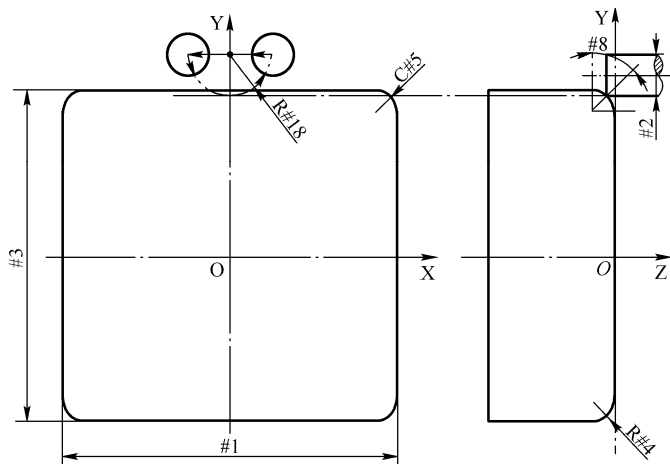


图 6-59 圆角外矩形倒圆加工

此例编程参照图 6-58 实例, 在它的基础上, 只要在宏程序的轮廓切削相关程

序段中,添加“ , R#5”即可。

这里应注意#5 要大于等于#4。另外,如果四周侧面圆角在上平面倒圆后要求半径值(#5)是不变,用固定半径补偿值编程方法编程时圆角半径#5 不变,而用 G10 指令格式编程时圆角半径#5 在倒圆时需加修正值 $\#4-\#4*\cos[\#8]$ 。如果要求半径值(#5)成规律递减,用 G10 指令格式编程更方便些,用固定半径补偿值编程方法编程时圆角半径#5 在倒圆时需减去修正值 $\#4-\#4*\cos[\#8]$ 。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 倒圆 R 值

#5=[J]; 圆角半径

#6=[K]; 递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

① 用 G10 指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒圆后从下向上递减宏程序。

%

O6433; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°

#26=#4-#4*SIN[#8]; 计算背吃刀量

#10=#2/2-#4+#4*COS[#8]; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器

G00 Z-#26; 快速下降到背吃刀量

G41 D1 X-#18; 半径补偿

G03 X0 Y[#3/2] R#18 F#7; 圆弧切入

G01 X[#1/2], R#5; 轮廓加工

Y-[#3/2], R#5; 轮廓加工

X-[#1/2], R#5; 轮廓加工

Y[#3/2], R#5; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#3/2+#18] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#8=#8+#6; 角度变量累加#6

END1; 循环 1 结束

M99; 子程序结束

%

② 用固定半径补偿值指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒圆后从下向上递减宏程序。

%

O4332; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°

#26=#4-#4*SIN[#8]; 计算背吃刀量

#20=#4*[1-COS[#8]]; 修正值

#15=#6-#20; 计算倒圆后圆角半径变化后的半径值

G00 Z-#26; 快速下降到背吃刀量

G41 D1 X-#18 Y[#3/2-#20+#18]; 半径补偿

G03 X0 Y[#3/2-#20] R#18 F#7; 圆弧切入

G01 X[#1/2-#20], #15; 轮廓加工

Y-[#3/2-#20], #15; 轮廓加工

X-[#1/2-#20], #15; 轮廓加工

Y[#3/2-#20], #15; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#3/2-#20] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#8=#8+#6; 角度变量累加#6

END1; 循环 1 结束

M99; 子程序结束

%

③ 用 G10 指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒圆后不变的宏程序。

%

O4333; 子程序号

```

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置
Z5; 快速下降到 Z=5 处
#8=0; 给角度变量#8 赋初始值 0
WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°
#26=#4-#4*SIN[#8]; 计算背吃刀量
#10=#2/2-#4+#4*COS[#8]; 计算半径补偿值#10
#15=#5+#4-#4*COS[#8]; 计算圆角半径在倒圆时的修正值
G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器
G00 Z-#26; 快速下降到背吃刀量
G41 D1 X-#18; 半径补偿
G03 X0 Y[#3/2] R#18 F#7; 圆弧切入
G01 X[#1/2], R#15; 轮廓加工
Y-[#3/2], R#15; 轮廓加工
X-[#1/2], R#15; 轮廓加工
Y[#3/2], R#15; 轮廓加工
X0; 轮廓加工
G03 X#18 Y[#3/2+#18] R#18; 圆弧切出
G40 G00 X0; 取消半径补偿
#8=#8+#6; 角度变量累加#6
END1; 循环 1 结束
M99; 子程序结束
%
```

④ 用固定半径补偿值指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒圆后不变的宏程序。

```

%
O4334; 子程序号
G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置
Z5; 快速下降到 Z=5 处
#8=0; 给角度变量#8 赋初始值 0
WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°
#26=#4-#4*SIN[#8]; 计算背吃刀量
#20=#4*[1-COS[#8]]; 修正值
G00 Z-#26; 快速下降到背吃刀量
G41 D1 X-#18 Y[#3/2-#20+#18]; 半径补偿
G03 X0 Y[#3/2-#20] R#18 F#7; 圆弧切入
```



```
G01 X[#1/2-#20], R#5; 轮廓加工
Y-[#3/2-#20], R#5; 轮廓加工
X-[#1/2-#20], R#5; 轮廓加工
Y[#3/2-#20], R#5; 轮廓加工
X0; 轮廓加工
G03 X#18 Y[#3/2-#20] R#18; 圆弧切出
G40 G00 X0; 取消半径补偿
#8=#8+#6; 角度变量累加#6
END1; 循环 1 结束
M99; 子程序结束
```

%

4) 主程序如下:

%

O433; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿

X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6433 A80 B16 C80 I5 J10 K10 D1000 R10; 调用宏程序 O6433,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

(2) 球头铣刀

在图 6-58 基础上, 在宏程序的轮廓切削相关程序段中, 添加 “, R#5” 即可。
程序略。

3. 倒角外矩形倒圆加工

(1) 立铣刀

1) 分析: 在图 6-60 基础上, 在宏程序的轮廓切削相关程序段中, 添加 “, C#5” 即可。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

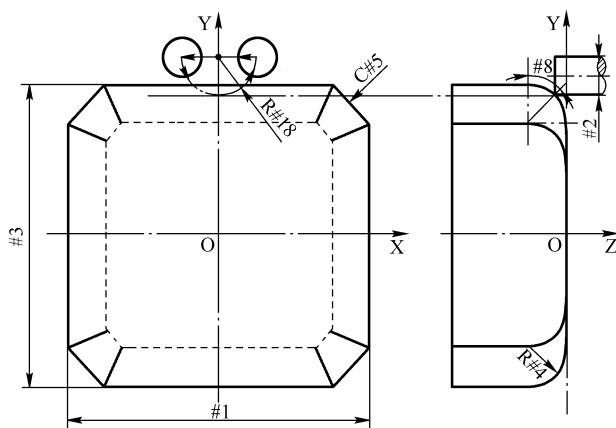


图 6-60 倒角外矩形倒圆加工

#4=[I]; 倒圆角 R 值

#5=[J]; 倒角 C 值

#6=[K]; 递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6434; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°

#26=#4-#4*SIN[#8]; 计算背吃刀量

#10=#2/2-#4+#4*COS[#8]; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器

G00 Z-#26; 快速下降到背吃刀量

G41 D1 X-#18; 半径补偿

G03 X0 Y[#3/2] R#18 F#7; 圆弧切入

G01 X[#1/2], C#5; 轮廓加工

Y-[#3/2], C#5; 轮廓加工

X-[#1/2], C#5; 轮廓加工

Y[#3/2], C#5; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#3/2+#18] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#8=#8+#6; 角度变量累加#6

END1; 循环 1 结束

M99; 子程序结束

%

4) 主程序如下:

%

O434; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿

X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6434 A80 B16 C80 I5 J10 K10 D1000 R10; 调用宏程序 O6434,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

(2) 球头铣刀

在图 6-58 基础上, 在宏程序的轮廓切削相关程序段中, 添加 “, C#5” 即可。
程序略。

4. 圆角内矩形倒圆加工

(1) 立铣刀

1) 分析: 如图 6-61 所示, 矩形槽长、宽分别为#1、#3, 矩形体四周倒圆半径为#5, 上平面四周倒圆半径为#4, 现在用直径为#2 的立铣刀进行倒圆加工。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 倒圆 R 值

#5=[J]; 圆角半径

#6=[K]; 递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

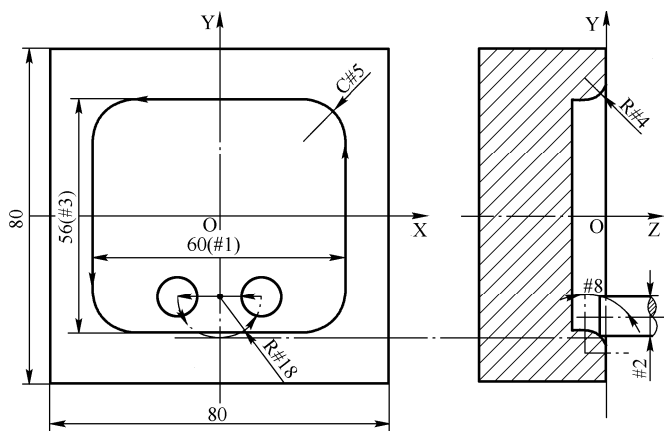


图 6-61 圆角内矩形倒圆加工

3) 宏程序如下:

① 用 G10 指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒圆后变化的宏程序。

%

O6435; 子程序号

G00 X0 Y[#18-#3/2]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°

#26=#4-#4*SIN[#8]; 计算背吃刀量

#10=#2/2-#4+#4*COS[#8]; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器

G00 Z-#26; 快速下降到背吃刀量

G41 D1 X-#18; 半径补偿

G03 X0 Y-[#3/2] R#18 F#7; 圆弧切入

G01 X[#1/2], R#5; 轮廓加工

Y[#3/2], R#5; 轮廓加工

X-[#1/2], R#5; 轮廓加工

Y-[#3/2], R#5; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#18-#3/2] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#8=#8+#6; 角度变量累加#6

END1; 循环 1 结束

M99; 子程序结束

%

② 用固定半径补偿值指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒圆后变化的宏程序。

%

O4352; 子程序号

G00 X0 Y[#18-#3/2]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°

#26=#4-#4*SIN[#8]; 计算背吃刀量

#20=#4*[1-COS[#8]]; 修正值

#15=#6-#20; 计算倒圆后圆角半径变化后的半径值

G00 Z-#26; 快速下降到背吃刀量

G41 D1 X-#18 Y[#3/2+#20+#18]; 半径补偿

G03 X0 Y[#3/2+#20] R#18 F#7; 圆弧切入

G01 X[#1/2+#20], #15; 轮廓加工

Y-[#3/2+#20], #15; 轮廓加工

X-[#1/2+#20], #15; 轮廓加工

Y[#3/2+#20], #15; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#3/2+#20] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#8=#8+#6; 角度变量累加#6

END1; 循环 1 结束

M99; 子程序结束

%

③ 用 G10 指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒圆后不变的宏程序。

%

O4353; 子程序号

G00 X0 Y[#18-#3/2]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

```

WHILE[#8LE90] DO1; 循环1 设定条件: 角度变量#8 不大于 90°
#26=#4-#4*SIN[#8]; 计算背吃刀量
#20=#4-#4*COS[#8]; 修正值
#10=#2/2-#20; 计算半径补偿值#10
#15=#5+#20; 计算圆角半径在倒圆时的修正值
G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器
G00 Z-#26; 快速下降到背吃刀量
G41 D1 X-#18; 半径补偿
G03 X0 Y-[#3/2] R#18 F#7; 圆弧切入
G01 X[#1/2], R#15; 轮廓加工
Y[#3/2], R#15; 轮廓加工
X-[#1/2], R#15; 轮廓加工
Y-[#3/2], R#15; 轮廓加工
X0; 轮廓加工
G03 X#18 Y[#18-#3/2] R#18; 圆弧切出
G40 G00 X0; 取消半径补偿
#8=#8+#6; 角度变量累加#6
END1; 循环1 结束
M99; 子程序结束
%
```

④ 用固定半径补偿值指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒圆后不变的宏程序。

```

%
O4354; 子程序号
G00 X0 Y[#18-#3/2]; 快速定位到加工准备位置
Z5; 快速下降到 Z=5 处
#8=0; 给角度变量#8 赋初始值 0
WHILE[#8LE90] DO1; 循环1 设定条件: 角度变量#8 不大于 90°
#26=#4-#4*SIN[#8]; 计算背吃刀量
#20=#4*[1-COS[#8]]; 修正值
G00 Z-#26; 快速下降到背吃刀量
G41 D1 X-#18 Y[#3/2+#20+#18]; 半径补偿
G03 X0 Y[#3/2+#20] R#18 F#7; 圆弧切入
G01 X[#1/2+#20], R#5; 轮廓加工
Y-[#3/2+#20], R#5; 轮廓加工
```

X-[#1/2+#20], R#5; 轮廓加工

Y[#3/2+#20], R#5; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#3/2+#20] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#8=#8+#6; 角度变量累加#6

END1; 循环 1 结束

M99; 子程序结束

%

4) 主程序如下:

%

O435; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿

X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6435 A80 B16 C80 I5 K10 D1000 R10; 调用宏程序 O6435,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

(2) 球头铣刀

略。

6.4.4 实例四矩形倒角加工

1. 直角外矩形倒角加工

1) 分析: 如图 6-62 所示, 矩形体长、宽分别为#1、#3, 上平面四周倒角为#4, 现在用直径为#2 的立铣刀进行倒角加工。

首先确定坐标系, 以工件上平面中心为坐标原点, 向右、向前、向上分别为 X、Y、Z 轴的正向。加工起始点从工件正前方, 中间点圆角与侧面切点(0, #3/2, -#4)开始圆弧切入, 切入圆弧半径为#18; 顺时针走刀路线, 完成一层切削后, 圆弧切出。为了编程坐标计算方便, 运用半径补偿功能, 直接用轮廓相关尺寸进行编程, 一层一层向上分层切削, 加工路线不变, 在这里以切削点与工件顶面距离为变量#26, 范围#4-0。

运用半径补偿就可利用轮廓尺寸进行编程,切削是从下往上,用 G10P1L12R#10 指令,就可以即时给半径补偿值指定变化后的数值,半径补偿值(D1)的变化规律: $\#10=\#2/2-[\#4-\#26]$, 这样编程坐标计算方便多了。

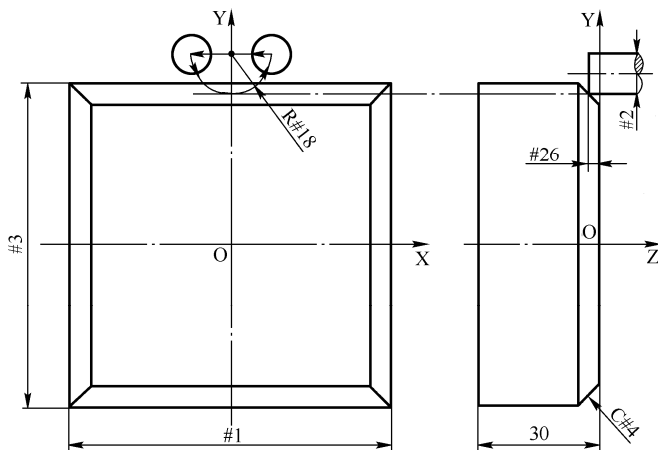


图 6-62 直角外矩形倒角加工

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 倒角 C 值

#6=[K]; 深度递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

#26=[Z]; 切削深度变量

3) 宏程序如下:

① 用 G10P1L12R#10 指令格式编写的宏程序。

%

O6441; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#26=#4; 给深度变量#26 赋初始值#4

WHILE[#26GE0] DO1; 循环 1 设定条件: 深度变量#26 不小于 0

#10=#2/2-[#4-#26]; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器


```
G00 Z-#26; 快速下降到切削深度
G41 D1 X-#18; 半径补偿
G03 X0 Y[#3/2] R#18 F#7; 圆弧切入
G01 X[#1/2]; 轮廓加工
Y-[#3/2]; 轮廓加工
X-[#1/2]; 轮廓加工
Y[#3/2]; 轮廓加工
X0; 轮廓加工
G03 X#18 Y[#3/2+#18] R#18; 圆弧切出
G40 G00 X0; 取消半径补偿
#26=#26-#6; 深度变量递减#6
END1; 循环 1 结束
M99; 子程序结束
%
```

② 用固定半径补偿值指令格式编写的宏程序。

```
%
O4412; 子程序号
G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置
Z5; 快速下降到 Z=5 处
#26=#4; 给深度变量#26 赋初始值#4
WHILE[#26GE0] DO1; 循环 1 设定条件: 切削深度变量#26≥0
G00 Z-#26; 快速下降到切削深度
#20=-#26+#4; 修正值
G41 D1 X-#18 Y[#3/2-#20+#18]; 半径补偿
G03 X0 Y[#3/2-#20] R#18 F#7; 圆弧切入
G01 X[#1/2-#20]; 轮廓加工
Y-[#3/2-#20]; 轮廓加工
X-[#1/2-#20]; 轮廓加工
Y[#3/2-#20]; 轮廓加工
X0; 轮廓加工
G03 X#18 Y[#3/2-#20] R#18; 圆弧切出
G40 G00 X0; 取消半径补偿
#26=#26-#6; 切削深度变量递减#6
END1; 循环 1 结束
M99; 子程序结束
```

%

4) 主程序如下:

%

O0441; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿
X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6441 A80 B16 C80 I5 K0.5 D1000 R10; 调用宏程序 O6441,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

2. 圆角外矩形倒角加工

1) 分析: 如图 6-63 所示, 矩形体长、宽分别为 #1、#3, 矩形体四周倒圆半径为 #5, 上平面四周倒角为 #4, 现在用直径为 #2 的立铣刀进行倒角加工。

此例编程参照图 6-62 实例, 在它的基础上, 只要在宏程序的轮廓切削相关程序段中, 添加 “, R#5” 即可。

这里应注意 #5 要大于等于 #4。另外, 如果四周侧面圆角在上平面倒圆后要求半径值 (#5) 是不变, 用固定半径补偿值编程方法编程时圆角半径 #5 不变, 而用 G10 指令格式编程时圆角半径 #5 在倒圆时需加修正值 $2 * \#4 - \#26$ 。如果要求半径值 (#5) 成规律递减, 用 G10 指令格式编程更方便些, 用固定半径补偿值编程方法编程时圆角半径 #5 在倒角时需减去修正值 $\#10 = \#2/2 - [\#4 - \#26]$ 。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 倒角 C 值

#5=[J]; 圆角半径

#6=[K]; 深度递增量

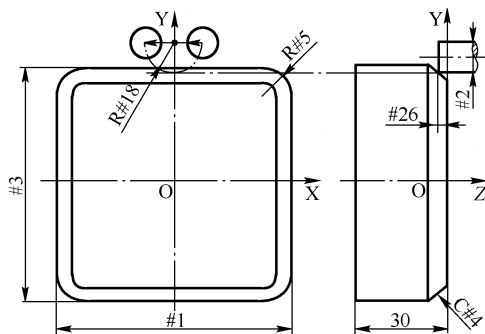


图 6-63 圆角外矩形倒角加工

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

#26=[Z]; 切削深度变量

3) 宏程序如下:

① 用 G10 指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒角后从下向上递减宏程序。

%

O6442; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#26=#4; 给深度变量#26 赋初始值#4

WHILE[#26GE0] DO1; 循环 1 设定条件: 深度变量#26 不小于 0

#10=#2/2-[#4-#26]; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器

G00 Z-#26; 快速下降到切削深度

G41 D1 X-#18; 半径补偿

G03 X0 Y[#3/2] R#18 F#7; 圆弧切入

G01 X[#1/2], R#5; 轮廓加工

Y-[#3/2], R#5; 轮廓加工

X-[#1/2], R#5; 轮廓加工

Y[#3/2], R#5; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#3/2+#18] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#26=#26-#6; 深度变量递减#6

END1; 循环 1 结束

M99; 子程序结束

%

② 用固定半径补偿值指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒角后从下向上递减宏程序。

%

O4422; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#26=#4; 给深度变量#26 赋初始值#4

```

WHILE[#26GE0] DO1; 循环1 设定条件: 深度变量#26 不小于 0
#24=#1/2-[#4-#26]; 计算 X 轴向极限尺寸
#25=#3/2-[#4-#26]; 计算 Y 轴向极限尺寸
#11=#5-[#4-#26]; 计算圆角半径修正值
G00 Z-#26; 快速下降到切削深度
G41 D1 X-#18 Y[#25+#18]; 半径补偿
G03 X0 Y#25 R#18 F#7; 圆弧切入
G01 X#24, R#11; 轮廓加工
Y-#25, R#11; 轮廓加工
X-#24, R#11; 轮廓加工
Y#25, R#11; 轮廓加工
X0; 轮廓加工
G03 X#18 Y[#25+#18] R#18; 圆弧切出
G40 G00 X0; 取消半径补偿
#26=#26-#6; 切削深度变量#26 递减一个深度增量#6
END1; 循环1 结束
M99; 子程序结束
%
```

③ 用 G10 指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒角后不变的宏程序。

```

%
O4423; 子程序号
G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置
Z5; 快速下降到 Z=5 处
#26=#4; 给深度变量#26 赋初始值#4
WHILE[#26GE0] DO1; 循环1 设定条件: 深度变量#26 不小于 0
#10=#2/2-[#4-#26]; 计算半径补偿值#10
#11=#5-[#4-#26]; 计算圆角半径修正值
G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器
G00 Z-#26; 快速下降到切削深度
G41 D1 X-#18; 半径补偿
G03 X0 Y[#3/2] R#18 F#7; 圆弧切入
G01 X#24, R#11; 轮廓加工
Y-#25, R#11; 轮廓加工
X-#24, R#11; 轮廓加工
```

Y#25, R#11; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#25+#18] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#26=#26-#6; 切削深度变量#26 递减一个深度增量#6

END1; 循环 1 结束

M99; 子程序结束

%

④ 用固定半径补偿值指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒圆后不变的宏程序。

%

O4424; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#26=#4; 给深度变量#26 赋初始值#4

WHILE[#26GE0] DO1; 循环 1 设定条件: 深度变量#26 不小于 0

#24=#1/2-[#4-#26]; 计算 X 轴向极限尺寸

#25=#3/2-[#4-#26]; 计算 Y 轴向极限尺寸

#11=#5-[#4-#26]; 计算圆角半径修正值

G00 Z-#26; 快速下降到切削深度

G41 D1 X-#18 Y[#25+#18]; 半径补偿

G03 X0 Y#25 R#18 F#7; 圆弧切入

G01 X#24, R#5; 轮廓加工

Y-#25, R#5; 轮廓加工

X-#24, R#5; 轮廓加工

Y#25, R#5; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#25+#18] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#26=#26-#6; 切削深度变量#26 递减一个深度增量#6

END1; 循环 1 结束

M99; 子程序结束

%

4) 主程序如下:

%

O442; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿
X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6442 A80 B16 C80 I5 J10 K0.5 D1000 R10; 调用宏程序 O6442,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

3. 倒角外矩形倒角加工 (图 6-64)

1) 分析: 在图 6-62 基础上, 在宏程序的轮廓切削相关程序段中, 添加 “, C#5” 即可。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 顶面四周倒角 C 值

#5=[J]; 侧面四角倒角 C 值

#6=[K]; 深度递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

#26=[Z]; 切削深度变量

3) 宏程序如下:

%

O6443; 子程序号

G00 X0 Y[#3/2+#18]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO1; 循环 1 设定条件: 角度变量#8 不大于 90°

#26=#4-#4*SIN[#8]; 计算切削深度

#10=#2/2-#4+#4*COS[#8]; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器

G00 Z-#26; 快速下降到切削深度

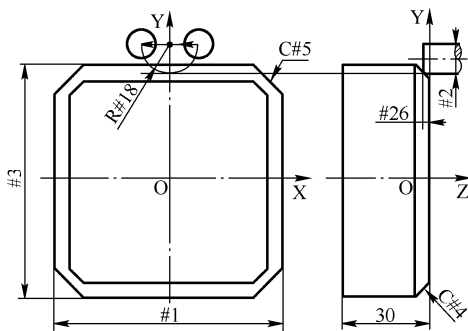


图 6-64 倒角外矩形倒角加工

```
G41 D1 X-#18; 半径补偿
G03 X0 Y[#3/2] R#18 F#7; 圆弧切入
G01 X[#1/2], C#5; 轮廓加工
Y-[#3/2], C#5; 轮廓加工
X-[#1/2], C#5; 轮廓加工
Y[#3/2], C#5; 轮廓加工
X0; 轮廓加工
G03 X#18 Y[#3/2+#18] R#18; 圆弧切出
G40 G00 X0; 取消半径补偿
#8=#8+#6; 角度变量累加#6
END1; 循环 1 结束
M99; 子程序结束
```

%

4) 主程序如下:

%

O443; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿

X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6443 A60 B16 C56 I5 J10 K0.5 D1000 R10; 调用宏程序 O6443,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

4. 圆角内矩形倒角加工

1) 分析: 如图 6-65 所示, 矩形槽长、宽分别为#1、#3, 矩形槽四周倒圆半径为#5, 上平面四周倒角为#4, 现在用直径为#2 的立铣刀进行倒角加工。

2) 变量及说明:

#1=[A]; 矩形长度

#2=[B]; 立铣刀直径

#3=[C]; 矩形宽度

#4=[I]; 倒角 C 值

#5=[J]; 圆角半径

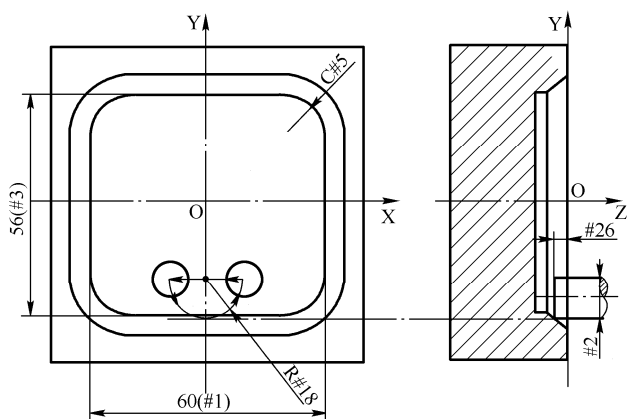


图 6-65 圆角内矩形倒角加工 (1)

#6=[K]; 深度递增量

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

#26=[Z]; 切削深度变量

3) 宏程序如下:

① 用 G10 指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒角时变化的宏程序。

%

O6444; 子程序号

G00 X0 Y[#18-#3/2]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#26=#4; 给深度变量#26 赋初始值#4

WHILE[#26GE0] DO1; 循环 1 设定条件: 深度变量#26 不小于 0

#10=#2/2-#4+#26; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器

G00 Z-#26; 快速下降到切削深度

G41 D1 X-#18; 半径补偿

G03 X0 Y-[#3/2] R#18 F#7; 圆弧切入

G01 X[#1/2], R#5; 轮廓加工

Y[#3/2], R#5; 轮廓加工

X-[#1/2], R#5; 轮廓加工

Y-[#3/2], R#5; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#18-#3/2] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#26=#26-#6; 切削深度变量递减#6

END1; 循环 1 结束

M99; 子程序结束

%

② 用固定半径补偿值指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒角时变化的宏程序。

%

O4442; 子程序号

G00 X0 Y[#18-#3/2]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#26=#4; 给深度变量#26 赋初始值#4

WHILE[#26GE0] DO1; 循环 1 设定条件: 深度变量#26 不小于 0

#24=#1/2+#4-#26; 计算 X 轴向极限尺寸

#25=#3/2+#4-#26; 计算 Y 轴向极限尺寸

#11=#5+#4-#26; 计算圆角半径修正值

G00 Z-#26; 快速下降到切削深量

G41 D1 X-#18 Y[#18-#25]; 半径补偿

G03 X0 Y-#25 R#18 F#7; 圆弧切入

G01 X[#24], R#11; 轮廓加工

Y[#25], R#11; 轮廓加工

X-[#24], R#11; 轮廓加工

Y-[#25], R#11; 轮廓加工

X0; 轮廓加工

G03 X#18 Y-[#18-#25] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#26=#26-#6; 切削深度变量#26 递减一个深度增量#6

END1; 循环 1 结束

M99; 子程序结束

%

③ 用 G10 指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒角时不变的宏程序 (图 6-66)。

%

O4443; 子程序号

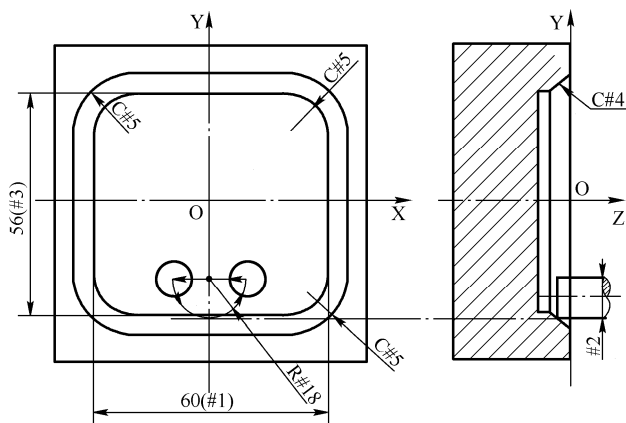


图 6-66 圆角内矩形倒角加工 (2)

```

G00 X0 Y[#18-#3/2]; 快速定位到加工准备位置
Z5; 快速下降到 Z=5 处
#26=#4; 给深度变量#26 赋初始值#4
WHILE[#26GE0] DO1; 循环 1 设定条件: 深度变量#26 不小于 0
#10=#2/2-#4+#26; 计算半径补偿值#10
G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器
#11=#5-#4+#26; 计算圆角半径修正值
G00 Z-#26; 快速下降到切削深度
G41 D1 X-#18; 半径补偿
G03 X0 Y-[#3/2] R#18 F#7; 圆弧切入
G01 X[#1/2] R#11; 轮廓加工
Y[#3/2] R#11; 轮廓加工
X-[#1/2] R#11; 轮廓加工
Y-[#3/2] R#11; 轮廓加工
X0; 轮廓加工
G03 X#18 Y[#18-#3/2] R#18; 圆弧切出
G40 G00 X0; 取消半径补偿
#26=#26-#6; 切削深度变量递减#6
END1; 循环 1 结束
M99; 子程序结束
%
```

④ 用固定半径补偿值指令格式编写的侧面四周圆角半径#5 在上平面轮廓倒角时不变的宏程序。

%

O4444; 子程序号

G00 X0 Y[#18-#3/2]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5 处

#26=#4; 给深度变量#26 赋初始值#4

WHILE[#26GE0] DO1; 循环 1 设定条件: 深度变量#26 不小于 0

#10=#2/2-#4+#26; 计算半径补偿值#10

G10 P1 L12 R#10; 将变量#10 值赋给半径补偿 D1 寄存器

G00 Z-#26; 快速下降到切削深度

G41 D1 X-#18; 半径补偿

G03 X0 Y-[#3/2] R#18 F#7; 圆弧切入

G01 X[#1/2] R#5; 轮廓加工

Y[#3/2] R#5; 轮廓加工

X-[#1/2] R#5; 轮廓加工

Y-[#3/2] R#5; 轮廓加工

X0; 轮廓加工

G03 X#18 Y[#18-#3/2] R#18; 圆弧切出

G40 G00 X0; 取消半径补偿

#26=#26-#6; 切削深度变量递减#6

END1; 循环 1 结束

M99; 子程序结束

%

4) 主程序如下:

%

O444; 主程序号

G54 G90 G00 G43 H1 Z50; 设定 G54 坐标系, 建立 H1 长度补偿

X0 Y0; 水平定位于 X0 Y0

M03 S1000; 主轴正转, 转速为 1000r/min

G65 P6444 A60 B16 C56 I5 J10 K0.3 D1000 R10; 调用宏程序 O6444,
并赋相关变量初始值

G00 G49 Z100; 取消长度补偿, 刀具抬高到 Z100

X0 Y150; 刀具返回到水平加工起始位 X0 Y150

M05; 主轴停止

M30; 程序结束, 返回程序头

%

6.5 倒角倒圆加工

6.5.1 实例一内外圆柱倒角

1. 外圆柱倒角

(1) 立铣刀

1) 分析: 如图 6-67 所示, 直径为 $\#1$ 的圆柱体上平面为 XY 轴所在的平面, 中心为坐标系原点, 轴线为 Z 轴, 上平面四周需用直径为 $\#2$ 的立铣刀进行倒角, C 为 $\#3$ 。

走刀方式: 下刀点即开始点选择在工件的右上角, 由下而上逐层环铣, 以 Z 轴坐标 $\#26$ 为自变量, $\#6$ 为递增量, 顺铣方式 (顺时针整圆铣削) 单向走刀。

首先快速定位在 $([\#1+\#2]/2+2, 0, -[\#3+2])$ 处, 赋自变量初始值 $\#26=\#3-\#6$, 计算此时刀具中心顶点 (参考点) 的 X 轴坐标 $\#24=[\#1+\#2]/2+\#26-\#3$, 直线切削到该点, 顺时针整圆铣削, 然后自变量递减一个递增量 $\#6$, 再计算此时刀具中心顶点的 X、Z 轴坐标, 直线切削到此点, 顺时针整圆铣削, 一层一层直至完成全部倒角切削加工。

2) 变量及说明:

$\#1=[A]$; 圆柱直径

$\#2=[B]$; 刀具直径

$\#3=[C]$; 倒角值

$\#6=[K]$; 深度递增量

$\#7=[D]$; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6511; 子程序号

$\#10=[\#1+\#2]/2$; 计算中心距

G00 X $[\#10+2]$ Y0; 快速定位到距工件右侧中心 2mm 处

Z $-\#3+2$; 快速下降到倒角下方 2mm 处

$\#26=\#3-\#6$; 计算倒角起始背吃刀量

WHILE $[\#26\text{GE}0]$ DO1; 循环 1 设定条件: 背吃刀量 $\#26$ 不小于 0

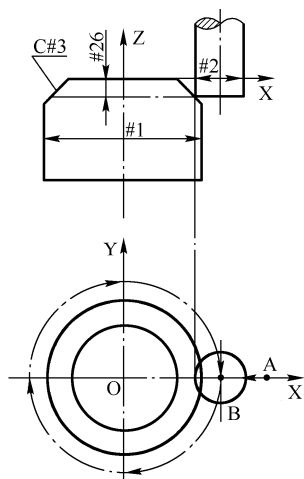


图 6-67 外圆柱倒角 (1)

#24=#10+#26-#3; 计算相应背吃刀量的 X 轴坐标值

G01 X#24 Z-#26 F#7; 直线切削到本轮切削起始点

G02 I-#24; 顺时针整圆圆弧切削

#26=#26-#6; 背吃刀量#26 递减一个深度增量#6

END1; 循环 1 结束

G00 Z50; 快速抬刀至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O511;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6511 A70 B16 C5 K0.5 D300; 调用宏程序 O6511, 给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

上述宏程序作为此类零件加工的模块子程序, 对于不同的直径、倒角尺寸, 只要在相应主程序调用指令 G65 中赋值即可。以下实例皆可如此调用。

(2) 圆鼻刀

1) 分析: 如图 6-68 所示, 与图 6-67 基本相同, 只不过刀具是刀尖有一个半径为#4 的圆鼻刀, 其余完全一样。

走刀方式: 与图 6-67 完全一样。

首先快速定位在 $([#1+#2]/2+2, 0, -[#3+2])$ 处, 赋自变量初始值 $\#11=\#3-\#6$, 计算此时刀具中心顶点 (参考点) 的 Z、X 轴坐标值分别为 $\#26=\#11+\#4*[1-\sin[45]]$, $\#24=\#10-\#3+\#11-\#4*[1-\cos[45]]$, 直线切削到该点, 顺时针整圆铣削,

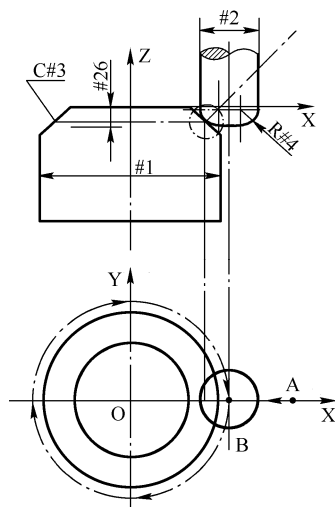


图 6-68 外圆柱倒角 (2)

然后自变量#11 上升一个递增量#6，再计算此时刀具中心顶点的 X、Z 轴坐标，直线切削到此点，顺时针整圆铣削，一层一层直至完成全部倒角切削加工。

2) 变量及说明：

#1=[A]；圆柱直径

#2=[B]；刀具直径

#3=[C]；倒角值

#4=[I]；刀尖角半径

#6=[K]；每次切削深度增量

#7=[D]；进给量，单位为 mm/min

3) 宏程序如下：

%

O6512；子程序号

#10=[#1+#2]/2；计算中心距

G00 X[#10+2] Y0；快速定位到距工件右侧中心 2mm 处

Z-[#3+2]；快速下降到倒角下方 2mm 处

#11=#3-#6；计算倒角起始背吃刀量

WHILE[#11GE0] DO1；循环 1 设定条件：背吃刀量#11 不小于 0

#26=#11+#4*[1-SIN[45]]；计算刀具中心顶点 Z 轴坐标值

#24=#10-#3+#11-#4*[1-COS[45]]；计算相应背吃刀量的 X 轴坐标值

G01 X#24 Z-#26 F#7；直线切削到本轮切削起始点

G02 I-#24；顺时针整圆圆弧切削

#11=#11-#6；背吃刀量#11 递减一个深度增量#6

END1；循环 1 结束

G00 Z50；快速抬刀至 Z=50

M99；子程序结束

%

4) 主程序如下：

%

O512；

G54 G90 G00 G43 H1 Z50；

X0 Y0；

M03 S1000；

G65 P6512 A70 B16 C5 I3 K0.5 D300；调用宏程序 O6512，给相关变量赋初始值

G00 G49 Z100；

X0 Y150;

M05;

M30;

%

(3) 球头铣刀

1) 分析: 如图 6-69 所示, 与图 6-67 基本相同, 只不过刀具是半径为 $\#2$ 的球头铣刀, 其余完全一样。

走刀方式: 与图 6-67 完全一样。

首先快速定位在 $(\#1/2+\#2*\text{SIN}[45]+2, 0, -[\#3-\#2*\text{SIN}[45]+2])$ 处, 赋自变量初始值 $\#11=\#3-\#6$, 计算此时刀具中心顶点 (参考点) 的 Z、X 轴坐标值分别为 $\#26=\#11+\#4*[1-\text{SIN}[45]]$, $\#24=\#1/2-\#3+\#10+\#2*\text{COS}[45]$, 直线切削到该点, 顺时针整圆铣削, 然后自变量 $\#11$ 上升一个递增量 $\#6$, 再计算此时刀具中心顶点的 X、Z 轴坐标, 直线切削到此点, 顺时针整圆铣削, 一层一层直至完成全部倒角切削加工。

2) 变量说明:

$\#1=[A]$; 圆柱直径

$\#2=[B]$; 球头铣刀半径

$\#3=[C]$; 倒角值

$\#6=[K]$; 深度递增量

$\#7=[D]$; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6513; 子程序号

G00 X $[\#1/2+\#2*\text{SIN}[45]+2]$ Y0; 快速定位到加工准备位置
Z $[-\#3-\#2*\text{SIN}[45]+2]$; 快速下降到距加工起始位置下方 2mm 处
 $\#11=\#3-\#6$; 设定倒角起始背吃刀量

WHILE $[\#11\text{GE}0]$ DO1; 循环 1 设定条件: 背吃刀量 $\#11$ 不小于 0

$\#26=\#11-\#2*\text{SIN}[45]$; 计算球头铣刀球心 Z 轴坐标值

$\#24=\#1/2-\#3+\#11+\#2*\text{COS}[45]$; 计算球头铣刀球心 X 轴坐标值

G01 X $\#24$ Z $\#26$ F $\#7$; 直线切削到本轮切削起始点

G02 I $\#24$; 顺时针整圆圆弧切削

$\#11=\#11-\#6$; 背吃刀量 $\#11$ 递减一个深度增量 $\#6$

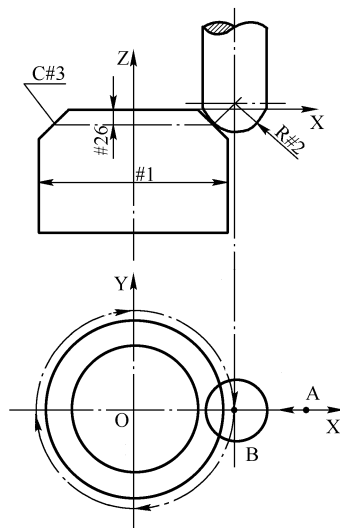


图 6-69 外圆柱倒角 (3)

END1; 循环 1 结束

G00 Z50; 快速抬刀至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O613;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6513 A70 B8 C5 I3 K0.5 D300; 调用宏程序 O6513, 给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 内圆柱倒角

(1) 立铣刀

1) 分析: 如图 6-70 所示, 内孔直径为#1 的圆柱体上平面为 XY 轴所在的平面, 中心为坐标系原点, 轴线为 Z 轴, 内孔孔口需用直径为#2 的立铣刀进行倒角, C 为#3。

走刀方式: 下刀点即开始点选择在工件孔口的右上角, 由下而上逐层环铣, 以 Z 轴坐标#26 为自变量, #6 为递增量, 顺铣方式 (逆时针整圆铣削) 单向走刀。

首先快速定位在 $([#1-#2]/2-2, 0, -[#3+2])$ 处, 赋自变量初始值 $\#26=\#3-\#6$, 计算此时刀具中心顶点 (参考点) 的 X 轴坐标 $\#24=[#1-#2]/2+\#26-\#3$, 直线切削到该点, 逆时针整圆铣削, 然后自变量上升一个递增量#6, 再计算此时刀具中心顶点的 X、Z 轴坐标, 直线切削到此点, 逆时针整圆铣削, 一层一层直至完成全部倒角切削加工。

2) 变量及说明:

#1=[A]; 圆柱直径

#2=[B]; 刀具直径

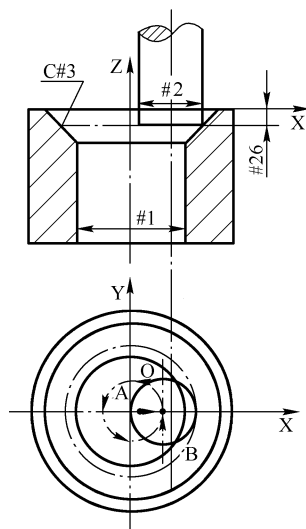


图 6-70 内圆柱倒角 (1)

#3=[C]; 倒角值

#6=[K]; 每次背吃刀量增量

#7=[D]; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6514; 子程序号

#10=[#1-#2]/2; 计算中心距

G00 X[#10-2] Y0; 快速定位到距工件右侧中心 2mm 处

Z-[#3+2]; 快速下降到倒角下方 2mm 处

#26=#3-#6; 设定倒角起始背吃刀量

WHILE[#26GE0] DO1; 循环 1 设定条件: 背吃刀量#26 不小于 0

#24=#10+#26-#3; 计算相应背吃刀量的 X 轴坐标值

G01 X#24 Z-#26 F#7; 直线切削到本轮切削起始点

G03 I-#24; 逆时针整圆圆弧切削

#26=#26-#6; 背吃刀量#26 递减一个深度增量#6

END1; 循环 1 结束

G00 Z50; 快速抬刀至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O514;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6514 A60 B16 C5 K0.5 D300; 调用宏程序 O6514, 给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 圆鼻刀

1) 分析: 如图 6-71 所示, 与图 6-70 基本相同, 只不过刀具是刀尖有一个半径为#4 的圆鼻刀, 其余完全一样。

走刀方式：与图 6-70 完全一样。

首先快速定位在 $(\#[1-\#2]/2-2, 0, -[\#3+2])$ 处，赋自变量初始值 $\#11=\#3-\#6$ ，计算此时刀具中心顶点（参考点）的 Z、X 轴坐标值分别为 $\#26=\#11+\#4*[1-\text{SIN}[45]]$ ， $\#24=\#10+\#3-\#11+\#4*[1-\text{COS}[45]]$ ，直线切削到该点，顺时针整圆铣削，然后自变量 $\#11$ 上升一个递增量 $\#6$ ，再计算此时刀具中心顶点的 X、Z 轴坐标，直线切削到此点，顺时针整圆铣削，一层一层直至完成全部倒角切削加工。

2) 变量及说明：

$\#1=[A]$ ；圆柱直径

$\#2=[B]$ ；刀具直径

$\#3=[C]$ ；倒角值

$\#4=[I]$ ；刀尖角半径

$\#6=[K]$ ；每次背吃刀量增量

$\#7=[D]$ ；进给量，单位为 mm/min

3) 宏程序如下：

%

O6515；子程序号

$\#10=[\#1-\#2]/2$ ；计算中心距

G00 X $\#[10-2]$ Y0；快速定位到距工件右侧中心 2mm 处

Z $-\#[3+2]$ ；快速下降到倒角下方 2mm 处

$\#11=\#3-\#6$ ；设定倒角起始背吃刀量

WHILE $[\#11\text{GE}0]$ DO1；循环 1 设定条件：背吃刀量 $\#11$ 不小于 0

$\#26=\#11+\#4*[1-\text{SIN}[45]]$ ；计算相应背吃刀量 $\#11$ 刀具中心顶点 X 轴坐标值

$\#24=\#10+\#3-\#11+\#4*[1-\text{COS}[45]]$ ；计算相应背吃刀量 $\#11$ 刀具中心顶点 Z 轴坐标值

G01 X $\#24$ Z $-\#26$ F $\#7$ ；直线切削到本轮切削起始点

G03 I $-\#24$ ；逆时针整圆圆弧切削

$\#11=\#11-\#6$ ；背吃刀量 $\#11$ 递减一个深度增量 $\#6$

END1；循环 1 结束

G00 Z50；快速抬刀至 Z=50

M99；子程序结束

%

4) 主程序如下：

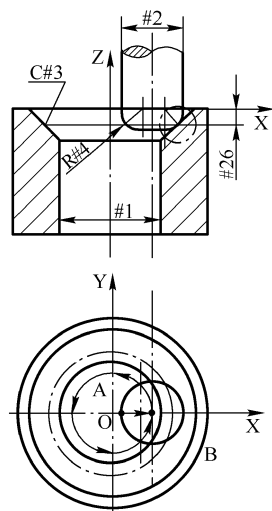


图 6-71 内圆柱倒角 (2)

```

%
O515;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6515 A60 B16 C5 I3 K0.5 D300; 调用宏程序 O6515, 给相
                                       关变量赋初始值

```

```

G00 G49 Z100;
X0 Y150;
M05;
M30;
%

```

(3) 球头铣刀

1) 分析: 如图 6-72 所示, 与图 6-70 基本相同, 只不过刀具是半径为 $\#2$ 的球头铣刀, 其余完全一样。

走刀方式: 与图 6-70 完全一样。

首先快速定位在 $(\#1/2 - \#2 * \sin[45] - 2, 0, -[\#3 - \#2 * \sin[45] + 2])$ 处, 赋自变量初始值 $\#11 = \#3 - \#6$, 计算此时刀具中心顶点 (参考点) 的 Z、X 轴坐标值分别为 $\#26 = \#11 - \#2 * [1 - \sin[45]]$, $\#24 = \#1/2 - \#2 * \cos[45] - \#11 + \#3$, 直线切削到该点, 顺时针整圆铣削, 然后自变量 $\#11$ 上升一个递增量 $\#6$, 再计算此时刀具中心顶点的 X、Z 轴坐标, 直线切削到此点, 顺时针整圆铣削, 一层一层直至完成全部倒角切削加工。

2) 变量说明:

$\#1 = [A]$; 圆柱直径

$\#2 = [B]$; 球头铣刀半径

$\#3 = [C]$; 倒角值

$\#6 = [K]$; 每次背吃刀量增量

$\#7 = [D]$; 进给量, 单位为 mm/min

3) 宏程序如下:

```

%

```

```

O6516; 子程序号

```

```

G00 X[#1/2 - #2 * COS[45] - 2] Y0; 快速定位到距工件右侧中心 2mm 处
Z-[#3 - #2 * SIN[45] + 2]; 快速下降到倒角下方 2mm 处

```

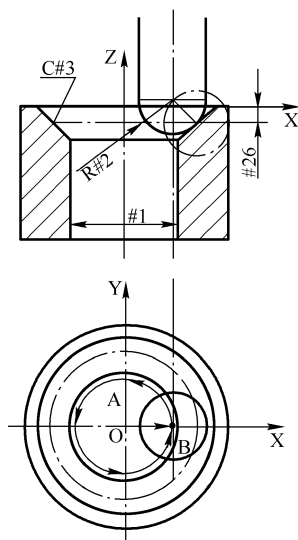


图 6-72 内圆柱倒角 (3)

#11=#3-#6; 设定倒角起始背吃刀量

WHILE[#11GE0] DO1; 循环1 设定条件: 背吃刀量#11 不小于 0

#24=#1/2-#2*COS[45]-#11+#3; 计算相应背吃刀量#11 刀具球心 X 坐标值

#26=#11-#2*SIN[45]; 计算相应背吃刀量#11 刀具球心 Z 坐标值

G01 X#24 Z-#26 F#7; 直线切削到本轮切削起始点

G03 I-#24; 逆时针整圆圆弧切削

#11=#11-#6; 背吃刀量递减一个深度增量#6

END1; 循环1 结束

G00 Z50; 快速抬刀至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O516;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6516 A60 B8 C5 K0.5 D300; 调用宏程序 O6516, 给相关变量
赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

6.5.2 实例二内外圆柱倒圆

1. 外圆柱倒圆

(1) 立铣刀

1) 分析: 如图 6-73 所示, 直径为#1 的圆柱体上平面为 X、Y 轴所在的平面, 中心为坐标系原点, 轴线为 Z 轴, 上平面四周需用直径为#2 的立铣刀进行倒圆, R 为#3。

走刀方式: 下刀点即开始点选择在工件的右上角, 由下而上逐层环铣, 以切削点跟倒圆圆心连线与水平面之间的夹角#5 为自变量, #6

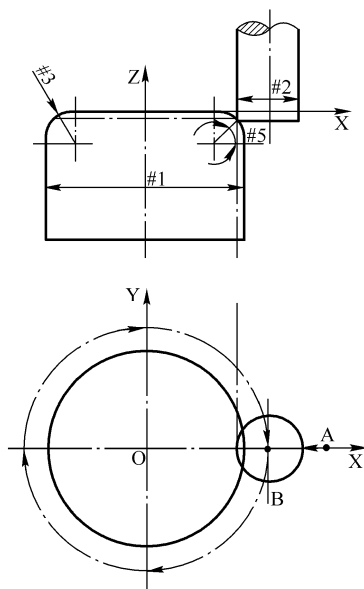


图 6-73 外圆柱倒圆 (1)

为角度递增量，顺铣方式（顺时针整圆铣削）单向走刀。

首先快速定位在 $([\#1+\#2]/2+2, 0, -[\#3+2])$ 处，赋自变量初始值 $\#5=\#6$ ，计算此时刀具中心顶点（参考点）的 X 轴坐标 $\#24=[\#1+\#2]/2-\#3*[1-\cos[\#5]]$ ，Z 轴坐标 $\#26=-\#3*[1-\sin[\#5]]$ ，直线切削到该点，顺时针整圆铣削，然后自变量上升一个角度递增量 $\#6$ ，再计算此时刀具中心顶点的 X、Z 轴坐标，直线切削到此点，顺时针整圆铣削，一层一层直至完成全部倒角切削加工。

2) 变量及说明：

$\#1=[A]$ ；圆柱直径

$\#2=[B]$ ；刀具直径

$\#3=[C]$ ；倒圆 $R=\#3$

$\#6=[K]$ ；每次角度增量

$\#7=[D]$ ；进给量，单位为 mm/min

3) 宏程序如下：

%

O6521；子程序号

$\#10=[\#1+\#2]/2$ ；计算中心距

G00 X $\#10+2$ Y0；快速定位到距工件右侧中心 2mm 处

Z- $\#3+2$ ；快速下降到倒角下方 2mm 处

$\#5=\#6$ ；设定倒圆起始切削角度为 $\#6$

WHILE $[\#5\leq 90]$ DO1；循环 1 设定条件：切削角度 $\#5$ 不大于 90°

$\#24=\#10-\#3*[1-\cos[\#5]]$ ；计算相应切削角度的 X 轴坐标值

$\#26=\#3*[1-\sin[\#5]]$ ；计算相应切削角度的 Z 轴坐标值

G01 X $\#24$ Z- $\#26$ F $\#7$ ；直线切削到本轮切削起始点

G02 I- $\#24$ ；顺时针整圆圆弧切削

$\#5=\#5+\#6$ ；切削角度递增一个角度增量 $\#6$

END1；循环 1 结束

G00 Z50；快速抬刀至 Z=50

M99；子程序结束

%

4) 主程序如下：

%

O621；

G54 G90 G00 G43 H1 Z50；

X0 Y0；

M03 S1000；

G65 P6521 A70 B16 C5 K5 D300; 调用宏程序 O6521, 给相关变量
赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 圆鼻刀

1) 分析: 如图 6-74 所示, 与图 6-73 基本相同, 只不过刀具是刀尖有一个半径为#4 的圆鼻刀, 其余完全一样。

走刀方式: 与图 6-73 完全一样。

首先快速定位在 $(\frac{[\#1+\#2]}{2}+2, 0, -[\#3+\#2])$ 处, 赋自变量初始值 $\#5=\#6$, 计算此时刀具中心顶点 (参考点) 的 X 轴坐标 $\#24=[\#1+\#2]/2-[\#3+\#4]*[1-\cos[\#5]]$, Z 轴坐标 $\#26=-[\#3+\#4]*[1-\sin[\#5]]$, 直线切削到该点, 顺时针整圆铣削, 然后自变量上升一个角度递增量#6, 再计算此时刀具中心顶点的 X、Z 轴坐标, 直线切削到此点, 顺时针整圆铣削, 一层一层直至完成全部倒角切削加工。

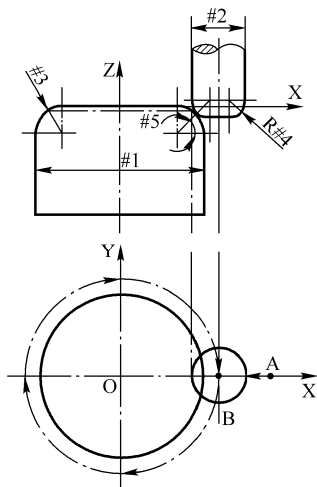


图 6-74 外圆柱倒圆 (2)

2) 变量及说明:

#1=[A]; 圆柱直径

#2=[B]; 刀具直径

#3=[C]; 倒圆 $R=\#3$

#4=[I]; 刀尖角半径

#6=[K]; 每次角度增量

#7=[D]; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6522; 子程序号

#10=[#1+#2]/2; 计算中心距

G00 X[#10+2] Y0; 快速定位到距工件右侧中心 2mm 处

Z-[\#3+\#2]; 快速下降到倒角下方 2mm 处

#5=#6; 设定倒圆起始切削角度为#6

WHILE[#5LE90] DO1; 循环 1 设定条件: 切削角度#5 不大于 90°

#24=#10-[\#3+\#4]*[1-COS[#5]]; 计算相应切削角度的 X 轴坐标值

#26=[#3+#4]*[1-SIN[#5]]; 计算相应切削角度的 Z 轴坐标值

G01 X#24 Z-#26 F#7; 直线切削到本轮切削起始点

G02 I-#24; 顺时针整圆圆弧切削

#5=#5+#6; 切削角度#5 递增一个角度增量#6

END1; 循环 1 结束

G00 Z50; 快速抬刀至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O522;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6522 A70 B16 C5 I3 K5 D300; 调用宏程序 O6522, 给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(3) 球头铣刀

1) 分析: 如图 6-75 所示, 与图 6-73 基本相同, 只不过刀具是半径为#2 的球头铣刀, 其余完全一样。

走刀方式: 与图 6-73 完全一样。

首先快速定位在 $([#1+#2]/2+2, 0, -[#3+2])$ 处, 赋自变量初始值#5=#6, 计算此时刀具中心顶点(参考点)的 X 轴坐标#24=#1/2-#3+[#2+#3]*COS[#5], Z 轴坐标#26=-#3+[#2+#3]*SIN[#5], 直线切削到该点, 顺时针整圆铣削, 然后自变量上升一个角度递增量#6, 再计算此时刀具中心顶点的 X、Z 轴坐标, 直线切削到此点, 顺时针整圆铣削, 一层一层直至完成全部倒角切削加工。

2) 变量及说明:

#1=[A]; 圆柱直径

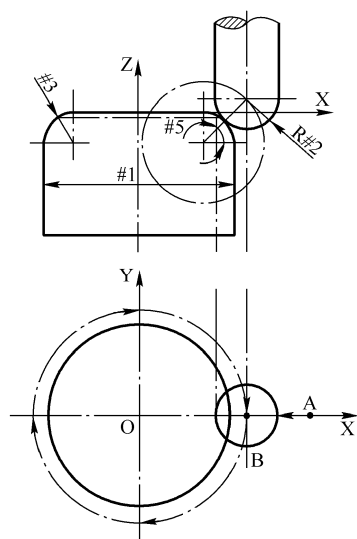


图 6-75 外圆柱倒圆 (3)

#2=[B]; 球头铣刀半径

#3=[C]; 倒圆 $R=\#3$

#6=[K]; 每次切削角度增量

#7=[D]; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6523; 子程序号

G00 X[#1/2+#3+2] Y0; 快速定位到加工准备位置

Z-[#3+2]; 快速下降到倒角下方 2mm 处

#5=#6; 设定切削角度初始值#6

WHILE[#5LE90] DO1; 循环 1 设定条件: 切削角度#5 不大于 90°

#24=#1/2-#3+[#2+#3]*COS[#5]; 计算相应切削角度#5 的 X 轴坐标值

#26=#3-[#2+#3]*SIN[#5]; 计算相应切削角度#5 的 Z 轴坐标值

G01 X#24 Z-#26 F#7; 直线切削到本轮切削起始点

G03 I-#24; 逆时针整圆圆弧切削

#5=#5+#6; 切削角度#5 递增一个角度增量#6

END1; 循环 1 结束

G00 Z50; 快速抬刀至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O523;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6523 A70 B8 C5 K10 D300; 调用宏程序 O6523, 给相关变量
赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

2. 内圆柱倒圆

(1) 立铣刀

1) 分析: 如图 6-76 所示, 内孔直径为#1 的圆柱体上平面为 X、Y 轴所在的平面, 中心为坐标系原点, 轴线为 Z 轴, 孔口四周需用直径为#2 的立铣刀进行倒圆, R 为#3。

走刀方式: 下刀点即开始点选择在工件孔口的右上角, 由下而上逐层环铣, 以切削点跟倒圆圆心连线与水平面之间的夹角#5 为自变量, #6 为角度递增量, 顺铣方式(顺时针整圆铣削)单向走刀。

首先快速定位在 $([#1-#2]/2-2, 0, -[#3+2])$ 处, 赋自变量初始值#5=#6, 计算此时刀具中心顶点(参考点)的 X 轴坐标#24= $[#1-#2]/2+#3*[1-\cos[#5]]$, Z 轴坐标#26= $-#3*[1-\sin[#5]]$, 直线切削到该点, 顺时针整圆铣削, 然后自变量上升一个角度递增量#6, 再计算此时刀具中心顶点的 X、Z 轴坐标, 直线切削到此点, 顺时针整圆铣削, 一层一层直至完成全部倒角切削加工。

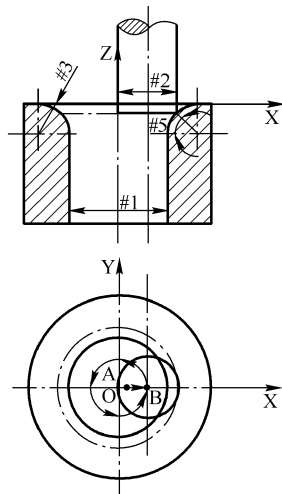


图 6-76 内圆柱倒圆 (1)

2) 变量及说明:

#1=[A]; 圆柱直径

#2=[B]; 刀具直径

#3=[C]; 倒圆 $R=#3$

#6=[K]; 每次角度增量

#7=[D]; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6524; 子程序号

#10=[#1-#2]/2; 计算中心距

G00 X[#10-2] Y0; 快速定位到距工件右侧中心 2mm 处

Z-[#3+2]; 快速下降到倒角下方 2mm 处

#5=#6; 设定倒圆起始角度

WHILE[#5LE90] DO1; 循环 1 设定条件: 切削角度#5 不大于 90°

#24=#10+#3*[1-COS[#5]]; 计算相应切削角度#5 的 X 轴坐标值

#26=#3*[1-SIN[#5]]; 计算相应切削角度#5 的 Z 轴坐标值

G01 X#24 Z-#26 F#7; 直线切削到本轮切削起始点

G03 I-#24; 逆时针整圆圆弧切削

#5=#5+#6; 切削角度#5 递增一个角度增量#6

END1; 循环 1 结束

G00 Z50; 快速抬刀至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O624;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6524 A60 B16 C5 K10 D300; 调用宏程序 O6524, 给相关变量
赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 圆鼻刀

1) 分析: 如图 6-77 所示, 与图 6-76 基本相同, 只不过刀具是刀尖有一个半径为 #4 的圆鼻刀, 其余完全一样。

走刀方式: 与图 6-76 完全一样。

首先快速定位在 $([#1-#2]/2-2, 0, -[#3+2])$ 处, 赋自变量初始值 $\#5=\#6$, 计算此时刀具中心顶点 (参考点) 的 X 轴坐标 $\#24=[#1-#2]/2+[#3+\#4]*[1-\cos[\#5]]$, Z 轴坐标 $\#26=-[#3+\#4]*[1-\sin[\#5]]$, 直线切削到该点, 顺时针整圆铣削, 然后自变量上升一个角度递增量 #6, 再计算此时刀具中心顶点的 X、Z 轴坐标, 直线切削到此点, 顺时针整圆铣削, 一层一层直至完成全部倒角切削加工。

2) 变量及说明:

#1=[A]; 圆柱直径

#2=[B]; 刀具直径

#3=[C]; 倒圆 $R=\#3$

#4=[I]; 刀尖角半径

#6=[K]; 每次角度增量

#7=[D]; 进给量, 单位为 mm/min

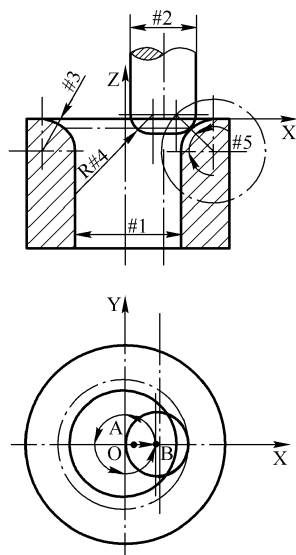


图 6-77 内圆柱倒圆 (2)

3) 宏程序如下:

%

O6525; 子程序号

#10=[#1-#2]/2; 计算中心距

G00 X[#10-2] Y0; 快速定位到距工件右侧中心 2mm 处

Z-[#3+2]; 快速下降到倒角下方 2mm 处

#5=#6; 设定倒圆起始角度

WHILE[#5LE90] DO1; 循环 1 设定条件: 切削角度#5 不大于 90°

#24=#10+[#3+#4]*[1-COS[#5]]; 计算相应切削角度#5 的 X 轴坐标值

#26=[#3+#4]*[1-SIN[#5]]; 计算相应切削角度#5 的 Z 轴坐标值

G01 X#24 Z-#26 F#7; 直线切削到本轮切削起始点

G03 I-#24; 逆时针整圆弧切削

#5=#5+#6; 切削角度#5 递增一个角度增量#6

END1; 循环 1 结束

G00 Z50; 快速抬刀至 Z=50

M99; 子程序结束

%

4) 主程序如下:

%

O525;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6525 A60 B16 C5 I3 K5 D300; 调用宏程序 O6525, 给相关
变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(3) 球头铣刀

1) 分析: 如图 6-78 所示, 与图 6-76 基本相同, 只不过刀具是半径为#2 的球头铣刀, 其余完全一样。

走刀方式: 与图 6-76 完全一样。

首先快速定位在 $([#1-#2]/2-2, 0, -[#3+2])$ 处, 赋自变量初始值#5=#6, 计

算此时刀具中心顶点（参考点）的 X 轴坐标 $\#24=\#1/2+\#3-[\#2+\#3]*\cos[\#5]$ ，Z 轴坐标 $\#26=-\#3+[\#2+\#3]*\sin[\#5]$ ，直线切削到该点，逆时针整圆铣削，然后自变量上升一个角度递增量 $\#6$ ，再计算此时刀具中心顶点的 X、Z 轴坐标，直线切削到此点，逆时针整圆铣削，一层一层直至完成全部倒角切削加工。

2) 变量及说明：

$\#1=[A]$ ；圆柱直径

$\#2=[B]$ ；球头铣刀半径

$\#3=[C]$ ；倒圆 $R=\#3$

$\#6=[K]$ ；角度增量

$\#7=[D]$ ；进给量，单位为 mm/min

3) 宏程序如下：

%

O6526；子程序号

G00 X $[\#1/2-\#2-2]$ Y0；快速定位到距工件右侧中心 2mm 处

Z $[-\#3+2]$ ；快速下降到倒角下方 2mm 处

$\#5=\#6$ ；设定倒圆起始角度

WHILE $[\#5\leq 90]$ DO1；循环 1 设定条件：切削角度 $\#5$ 不大于 90°

$\#24=\#1/2+\#3-[\#2+\#3]*\cos[\#5]$ ；计算相应切削角度 $\#5$ 的 X 轴坐标值

$\#26=\#3-[\#2+\#3]*\sin[\#5]$ ；计算相应切削角度 $\#5$ 的 Z 轴坐标值

G01 X $\#24$ Z $-\#26$ F $\#7$ ；直线切削到本轮切削起始点

G03 I $-\#24$ ；逆时针整圆圆弧切削

$\#5=\#5+\#6$ ；切削角度 $\#5$ 递增一个角度增量 $\#6$

END1；循环 1 结束

G00 Z50；快速抬刀至 Z=50

M99；子程序结束

%

4) 主程序如下：

%

O626；

G54 G90 G00 G43 H1 Z50；

X0 Y0；

M03 S1000；

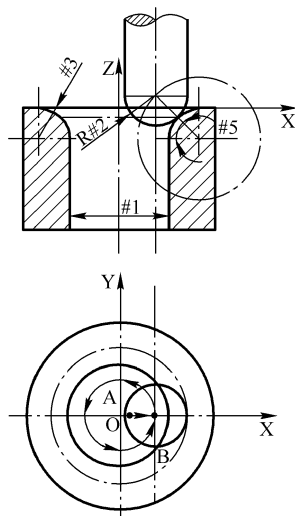


图 6-78 内圆柱倒圆 (3)

G65 P6526 A60 B8 C5 K5 D300; 调用宏程序 O6526, 给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

6.5.3 实例三内外形倒角

1. 外形倒角

1) 分析: 如图 6-79 所示, 以工件的上平面为 X、Y 轴所在的平面, 中心为坐标系原点, Z 轴垂直于上平面, 上平面四周需用直径为 #2 的立铣刀进行倒角, C 为 #3。

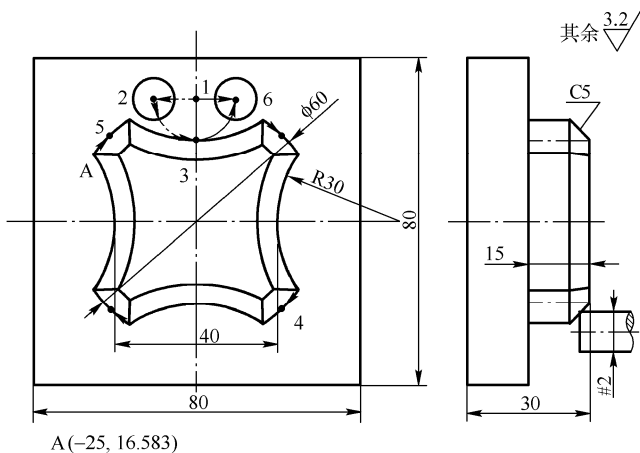


图 6-79 外形倒角

走刀方式: 由于工件形状不具同性, 因此不能作为需赋初始值的宏程序, 只作特定零件的加工程序, 具体零件轮廓基点坐标不同, 其他同性的可作参考。下刀点即开始点选择在工件的正前方, 采用圆弧切入切出, 圆弧半径设为 #18, 可根据具体零件大小等不同, 赋不同的数值。由下而上逐层环铣, 以 Z 轴坐标 #26 为自变量, #6 为递增量, 顺铣方式 (顺时针整圆铣削) 单向走刀。这里还有一个关键的因素需要考虑: 半径补偿及补偿值确定, 刚开始切削补偿值为刀具半径, 随着不断上升, 各基点坐标不变, 半径补偿值就需要相应变化, 它的变化规律是: 倒角是 45°, 从开始加工深度归零, 刀具上升多少, 半径补偿值就减少多少, 这可以运用 G10 L12 PΔR[#2/2-#3+#26](PΔ为半径补偿值 D 代码, 如: P1 即为 D1)。

首先快速定位在 (0, [20+#18], -#3) 处, 赋自变量初始值 #26=#3, 通过半

径补偿,再加工圆弧切入切出,就可直接用零件轮廓相关基点坐标进行编程,顺时针走刀铣削,然后自变量上升一个递增量#6,再计算此时的刀具半径补偿值 $\#1=\#2/2-\#3+\#26$,抬刀至相应背吃刀量#26,接着轮廓铣削,一层一层直至完成全部倒角切削加工。

2) 变量及说明:

#2=[B]; 刀具直径

#3=[C]; 倒角 C 值

#6=[K]; 深度增量

#7=[D]; 进给量,单位为 mm/min

#18=[R]; 1/4 圆弧切入切出半径

3) 加工程序如下:

%

O6531; 程序号

G54 G90 G43 H1 G00 Z50; 快速下刀到 Z=50 处,完成长度补偿

X0 Y0; 快速定位到 (0, 0)

M03 S1000; 主轴正转,转速为 1000r/min

#2=16; 给相关变量赋初始值

#3=5; 给相关变量赋初始值

#6=0.2; 给相关变量赋初始值

#7=1000; 给相关变量赋初始值

#18=10; 给相关变量赋初始值

G00 X0 Y[20+#18]; 快速定位到加工准备位置

Z-#3; 快速下降到加工起始背吃刀量

#26=#3; 给背吃刀量变量赋初始值#3

WHILE[#26GE0] DO1; 循环 1 设定条件: 背吃刀量#26 不小于 0

G01 Z-#26 F#7; 直线切削至背吃刀量#26

#1=#2/2-#3+#26; 计算对应于背吃刀量#26 的半径补偿值#1

G10 P1 L12 R#1; 将半径补偿值#1 赋给 D1

G41 D1 G00 X-#18; 建立半径补偿

G03 X0 Y20 R#18; 圆弧切入

X16.583 Y25 R30; 轮廓加工

G02 X25 Y16.583 R30; 轮廓加工

G03 Y-16.385 R30; 轮廓加工

G02 X16.385 Y-25 R30; 轮廓加工

G03 X-16.385 R30; 轮廓加工

```

G02 X-25 Y-16.385 R30; 轮廓加工
G03 Y16.385 R30; 轮廓加工
G02 X-16.385 Y25 R30; 轮廓加工
G03 X0 Y20 R30; 轮廓加工
X#18 Y[20+#18] R#18; 圆弧切出
G40 G00 X0; 取消半径补偿
#26=#26-#6; 背吃刀量#26 递减一个深度递增量#6
END1; 循环 1 结束
G49 G00 Z100; 快速抬刀到 Z=100 处, 并取消长度补偿
X0 Y150; 快速返回程序运行起始点
M05; 主轴停止
M30; 程序结束, 返回程序头
%
```

2. 内腔倒角

1) 分析: 如图 6-80 所示, 以工件的上平面为 XY 轴所在的平面, 中心为坐标系原点, Z 轴垂直于上平面, 内腔孔口需用直径为 $\phi 2$ 的立铣刀进行倒角, C 为 $\phi 3$ 。

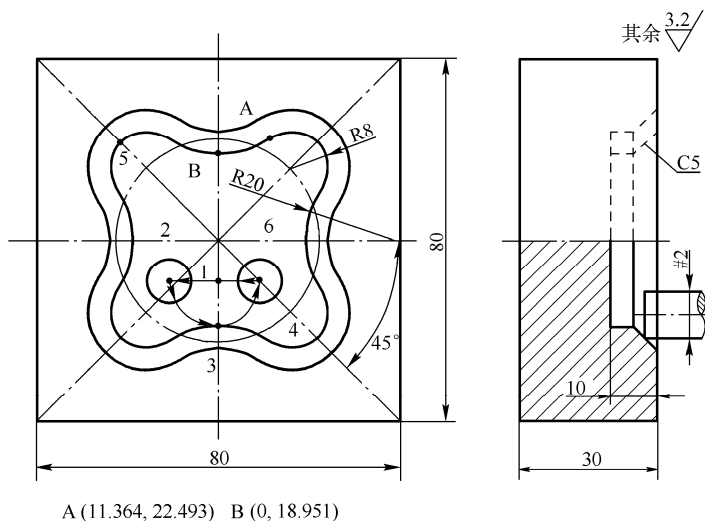


图 6-80 内腔倒角

走刀方式: 下刀点即开始点选择在内腔的正前方 (0, -20), 采用圆弧切入切出, 圆弧半径设为 $\phi 18$, 可根据具体零件大小等不同, 赋不同的数值。由下而上逐层环铣, 以 Z 轴坐标 $\phi 26$ 为自变量, $\phi 6$ 为递增量, 顺铣方式 (逆时针绕内腔轮廓铣削) 单向走刀。这里还有一个关键的因素需要考虑: 半径补偿及补偿值确定, 刚开始切削补偿值为刀具半径, 随着不断上升, 各基点坐标不变, 半径补偿值 ($\phi 1$)

就需要相应变化，它的变化规律是：倒角是 45° ，从开始加工半径补偿值 $\#1=\#2/2$ ，刀具上升多少，半径补偿值 $\#1$ 就减少多少，这可以运用 G10 L12 P1 R $\#1$ 。

首先快速定位在 $(0, -[20-\#18], -\#3)$ 处，赋自变量初始值 $\#26=\#3$ ，通过半径补偿，再加工圆弧切入切出，就可直接用零件轮廓相关基点坐标进行编程，逆时针走刀铣削，然后自变量上升一个递增量 $\#6$ ，再计算此时的刀具半径补偿值 $\#1=\#2/2-\#3+\#26$ ，抬刀至相应背吃刀量 $\#26$ ，接着轮廓铣削，一层一层直至完成全部倒角切削加工。

2) 变量及说明：

$\#2=[B]$ ；刀具直径

$\#3=[C]$ ；倒角 C 值

$\#6=[K]$ ；深度增量

$\#7=[D]$ ；进给量，单位为 mm/min

$\#18=[R]$ ；1/4 圆弧切入切出半径

3) 加工程序如下：

%

O6532；程序号

G54 G90 G43 H1 G00 Z50；快速下刀到 Z=50 处，完成长度补偿

X0 Y0；快速定位到 $(0, 0)$

M03 S1000；主轴正转，转速为 1000r/min

$\#2=16$ ；给相关变量赋初始值

$\#3=5$ ；给相关变量赋初始值

$\#6=0.2$ ；给相关变量赋初始值

$\#7=1000$ ；给相关变量赋初始值

$\#18=10$ ；给相关变量赋初始值

G00 X0 Y $[-18.951+\#18]$ ；快速定位到加工准备位置

Z $-\#3$ ；快速下降到加工起始背吃刀量

$\#26=\#3$ ；给背吃刀量变量赋初始值 $\#3$

WHILE $[\#26GE0]$ DO1；循环 1 设定条件：背吃刀量 $\#26$ 不小于 0

G01 Z $-\#26$ F $\#7$ ；直线切削至背吃刀量 $\#26$

$\#1=\#2/2-\#3+\#26$ ；计算对应于背吃刀量 $\#26$ 的半径补偿值 $\#1$

G10 P1 L12 R $\#1$ ；将半径补偿值 $\#1$ 赋给 D1

G41 D1 G00 X $-\#18$ ；建立半径补偿

G03 X0 Y -18.951 R $\#18$ ；圆弧切入

G02 X11.364 Y -22.493 R20；轮廓加工

G03 X22.493 Y -11.364 R8；轮廓加工


```

G02 Y11.364 R20; 轮廓加工
G03 X11.364 Y22.493 R8; 轮廓加工
G02 X-11.364 R20; 轮廓加工
G03 X-22.493 Y11.364 R8; 轮廓加工
G02 Y-11.364 R20; 轮廓加工
G03 X-11.364 Y-22.493 R8; 轮廓加工
G02 X0 Y-18.951 R20; 轮廓加工
G03 X#18 Y[-18.951+#18] R#18; 圆弧切出
#26=#26-#6; 背吃刀量#26 递减一个深度递增量#6
END1; 循环 1 结束
G49 G00 Z100; 快速抬刀到 Z=100 处, 并取消长度补偿
X0 Y150; 快速返回程序运行起始点
M05; 主轴停止
M30; 程序结束, 返回程序头
%
```

6.5.4 实例四内外形倒圆

1. 外形倒圆

(1) 立铣刀

1) 分析: 如图 6-81 所示, 以工件的上平面为 XY 轴所在的平面, 中心为坐标系原点, Z 轴垂直于上平面, 上平面四周需用直径为 #2 的立铣刀进行倒圆, r 为 #3。

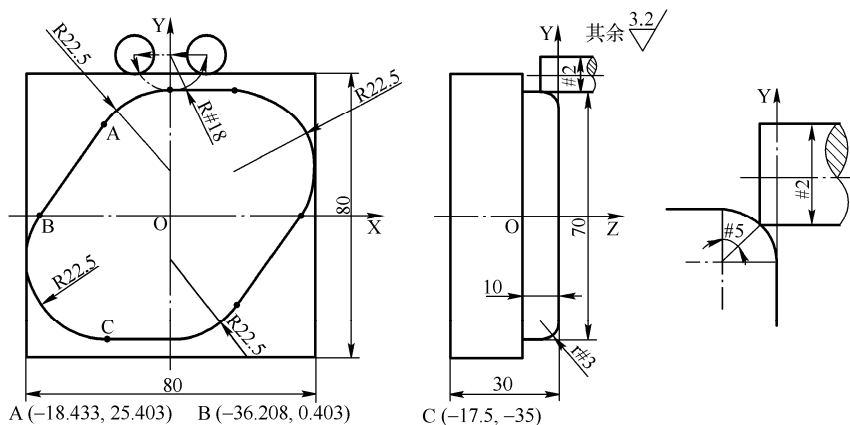


图 6-81 外形倒圆 (1)

走刀方式: 下刀点即开始点选择在工件的正前方, 采用圆弧切入切出, 圆弧

半径设为#18, 可根据具体零件大小等不同, 赋不同的数值。由下而上逐层环铣, 以切削点与圆角圆心连线和水平面间夹角#5 为自变量, #6 为角递增量, 顺铣方式 (顺时针沿零件外轮廓铣削) 单向走刀。开始加工#5 归零, 随着刀具上升#5 按照#6 递增, 半径补偿值就按照 $\#1=\#2/2-\#3*[1-\cos[\#5]]$ 规律减少, 背吃刀量#26 按照 $\#26=\#3*[1-\sin[\#5]]$ 变化。这可以运用 G10 L12 P1 R#1 进行赋值给 D1 寄存器, 从而一直以零件外轮廓相关基点坐标编程, 由半径补偿值的变化、背吃刀量的变化进行圆角加工。

首先快速定位在 (0, #18+35, -#3) 处, 赋角度量初始值#5=0 (通过半径补偿), 再加工圆弧切入切出, 就可直接用零件轮廓相关基点坐标进行编程, 顺时针走刀铣削, 然后自变量增加一个角度递增量#6, 再计算此时的刀具半径补偿值#1 和背吃刀量#26, 抬刀至相应背吃刀量#26, 接着轮廓铣削, 一层一层直至完成全部圆角切削加工。

2) 变量及说明:

#2=[B]; 刀具直径

#3=[C]; 倒圆半径

#6=[K]; 角度增量

#7=[D]; 进给量, 单位为 mm/min

#18=[R]; 1/4 圆弧切入切出半径

3) 加工程序如下:

%

O6541; 程序号

G54 G90 G43 H1 G00 Z50; 快速下刀到 Z=50 处, 完成长度补偿

X0 Y0; 快速定位到 (0, 0)

M03 S1000; 主轴正转, 转速为 1000r/min

#2=16; 给相关变量赋初始值

#3=5; 给相关变量赋初始值

#6=5; 给相关变量赋初始值

#7=1000; 给相关变量赋初始值

#18=10; 给相关变量赋初始值

X0 Y[#18+35]; 快速定位到加工准备位置

Z5; 快速定位到加工准备深度位置

#5=0; 给角度变量#5 赋初始值 0

WHILE[#5LE90] DO1; 循环 1 设定条件: 角度变量#5 不大于 90°

#26=#3*[1-SIN[#5]]; 计算相对于角度变量#5 的背吃刀量#26

#1=#2/2-#3*[1-COS[#5]]; 计算对应于角度变量#5 的半径补偿值#1

```

G10 P1 L12 R#1; 将半径补偿值#1 赋给 D1
G41 G00 D1 X-#18; 建立半径补偿
G01 Z-#26 F100; 直线切削到背吃刀量#26
G03 X0 Y35 R#18 F#7; 圆弧切入
G01 X17.5; 轮廓加工
G02 X36.208 Y-0.403 R22.5; 轮廓加工
G01 X18.433 Y-26.403; 轮廓加工
G02 X0 Y-35 R22.5; 轮廓加工
G01 X-17.5; 轮廓加工
G02 X-36.208 Y0.403 R22.5; 轮廓加工
G01 X-18.433 Y26.403; 轮廓加工
G02 X0 Y35 R22.5; 轮廓加工
G03 X#18 Y[#18+35] Z[2-#26] R#18; 圆弧切出
G40 G0 X0; 取消半径补偿
#5=#5+#6; 角度变量 6 递增一个角度递增量#6
END1; 循环 1 结束
G49 G00 Z100; 快速抬刀到 Z=100 处, 并取消长度补偿
X0 Y150; 快速返回程序运行起始点
M05; 主轴停止
M30; 程序结束, 返回程序头
%
```

(2) 球头铣刀

1) 分析: 如图 6-82 所示, 可参照图 6-81 分析, 只是刀具为半径为#2 的球头铣刀, 参考点为球心, 坐标计算有所不同, 方法相同。

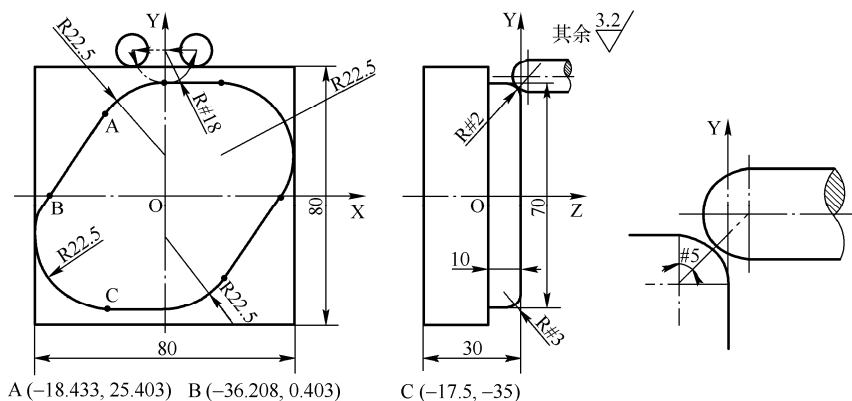


图 6-82 外形倒圆 (2)

2) 变量及说明:

#2=[B]; 球头铣刀半径

#3=[C]; 倒圆半径

#6=[K]; 角度增量

#7=[D]; 进给量, 单位为 mm/min

#18=[R]; 1/4 圆弧切入切出半径

3) 加工程序如下:

%

O6542; 程序号

G54 G90 G43 H1 G00 Z50; 快速下刀到 Z=50 处, 完成长度补偿

X0 Y0; 快速定位到 (0, 0)

M03 S1000; 主轴正转, 转速为 1000r/min

#2=16; 给相关变量赋初始值

#3=5; 给相关变量赋初始值

#6=5; 给相关变量赋初始值

#7=1000; 给相关变量赋初始值

#18=10; 给相关变量赋初始值

X0 Y[#18+35]; 快速定位到加工准备位置

Z5 ; 快速定位到加工准备深度位置

#5=0; 给角度变量#5 赋初始值 0

WHILE[#5LE90] DO1; 循环 1 设定条件: 角度变量#5 不大于 90°

#26=[#2+#3]*[1-SIN[#5]]; 计算相对于角度变量#5 的背吃刀量#26

#1=-[#2+#3]*[1-COS[#5]]; 计算对应于角度变量#5 的半径补偿值#1

G10 P1 L12 R#1; 将半径补偿值#1 赋给 D1

G41 G00 D1 X-#18; 建立半径补偿

G01 Z-#26 F[0.3*#7]; 直线切削至背吃刀量#26

G03 X0 Y35 R#18 F#7; 圆弧切入

G01 X17.5; 轮廓加工

G02 X36.208 Y-0.403 R22.5; 轮廓加工

G01 X18.433 Y-26.403; 轮廓加工

G02 X0 Y-35 R22.5; 轮廓加工

G01 X-17.5; 轮廓加工

G02 X-36.208 Y0.403 R22.5; 轮廓加工

G01 X-18.433 Y26.403; 轮廓加工

G02 X0 Y35 R22.5; 轮廓加工

```

G03 X#18 Y[#18+35] Z[2-#26] R#18; 圆弧切出
G40 G0 X0; 取消半径补偿
#5=#5+#6; 角度变量#5 递增一个角度递增量#6
END1; 循环 1 结束
G49 G00 Z100; 快速抬刀到 Z=100 处, 并取消长度补偿
X0 Y150; 快速返回程序运行起始点
M05; 主轴停止
M30; 程序结束, 返回程序头
%
```

2. 内腔倒圆

1) 分析: 如图 6-83 所示, 以工件的上平面为 XY 轴所在的平面, 中心为坐标系原点, Z 轴垂直于上平面, 内腔孔口用直径为 #2 的立铣刀进行倒圆, R 为 #3。

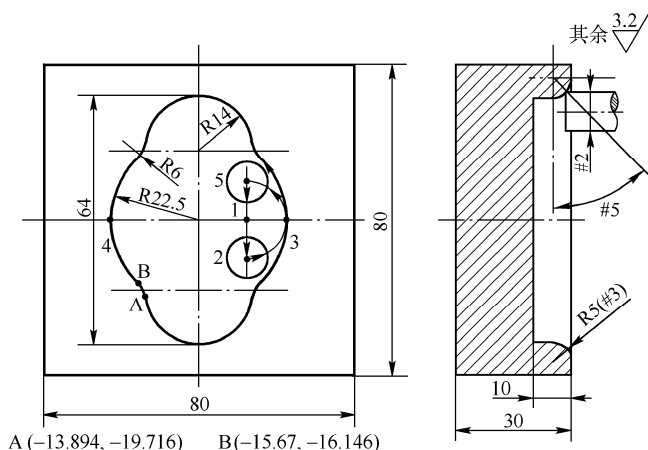


图 6-83 内腔倒圆

走刀方式: 下刀点即开始点选择在工件的正前方, 采用圆弧切入切出, 圆弧半径设为 #18, 可根据具体零件大小等不同, 赋不同的数值。由下而上逐层环铣, 以切削点与圆角圆心连线和水平面间夹角 #5 为自变量, #6 为角递增量, 顺铣方式 (逆时针沿内腔轮廓铣削) 单向走刀。开始加工 #5 归零, 随着刀具上升 #5 按照 #6 递增, 半径补偿值就按照 $\#1=\#2/2-\#3*\cos[\#5]$ 规律减少, 背吃刀量 #26 按照 $\#26=\#3-\#3*\sin[\#5]$ 变化。这可以运用 G10L12P1R#1 进行赋值给 D1 寄存器, 从而一直以零件外轮廓相关基点坐标编程, 由半径补偿值的变化、背吃刀量的变化进行圆角加工。

首先快速定位在 $(22.6-\#18, 0, -\#3)$ 处, 赋角度量初始值 #5=0 通过半径补偿, 再加工圆弧切入切出, 就可直接用零件轮廓相关基点坐标进行编程, 顺时针

走刀铣削，然后自变量增加一个角度递增量#6，再计算此时的刀具半径补偿值#1和背吃刀量#26，抬刀至相应背吃刀量#26，接着轮廓铣削，一层一层直至完成全部圆角切削加工。

2) 变量及说明：

#2=[B]；刀具直径

#3=[C]；倒圆半径

#6=[K]；角度增量

#7=[D]；进给量，单位为 mm/min

#18=[R]；1/4 圆弧切入切出半径

3) 加工程序如下：

%

O6543；程序号

G54 G90 G43 H1 G00 Z50；快速下刀到 Z=50 处，完成长度补偿

X0 Y0；快速定位到 (0, 0)

M03 S1000；主轴正转，转速为 1000r/min

#2=16；给相关变量赋初始值

#3=5；给相关变量赋初始值

#6=5；给相关变量赋初始值

#7=1000；给相关变量赋初始值

#18=10；给相关变量赋初始值

X[22.6-#18] Y0；快速定位到加工准备位置

Z5；快速定位到加工准备深度位置

#5=0；给角度变量#5 赋初始值 0

WHILE[#5LE90] DO1；循环 1 设定条件：角度变量#5 不大于 90°

#26=#3*[1-SIN[#5]]；计算相对于角度变量#5 的背吃刀量#26

#1=#2/2-#3*COS[#5]；计算对应于角度变量#5 的半径补偿值#1

G10 P1 L12 R#1；将半径补偿值#4 赋给 D1

G41 G00 D1 Y-#18；建立半径补偿

G00 Z-#26；快速下降到背吃刀量#26

G03 X22.5 Y0 R#18 F#7；圆弧切入

G03 X16.67 Y16.146 R22.5；轮廓加工

G02 X13.894 Y19.716 R6；轮廓加工

G03 X-13.894 R14；轮廓加工

G02 X-16.67 Y16.146 R6；轮廓加工

G03 Y-16.146 R22.5；轮廓加工

```

G02 X-13.894 Y-19.716 R6; 轮廓加工
G03 X13.894 R14; 轮廓加工
G02 X16.67 Y-16.146 R6; 轮廓加工
G03 X22.5 Y0 R22.5; 轮廓加工
G03 X[22.6-#18] Y#18 R#18; 圆弧切出
G40 G0 X0; 取消半径补偿
#5=#5+#6; 角度变量#5 递增一个角度递增量#6
END1; 循环 1 结束
G49 G00 Z100; 快速抬刀到 Z=100 处, 并取消长度补偿
X0 Y150; 快速返回程序运行起始点
M05; 主轴停止
M30; 程序结束, 返回程序头
%
```

6.5.5 实例五斜面加工

1. 斜面粗加工

1) 分析: 建立如图 6-84 所示坐标系, 采用从上而下、每层厚度为#6、由右以行间距#8=0.8*#2 向中心逐次铣削, 先加工右半斜面, 再利用镜像指令, 加工左半斜面。

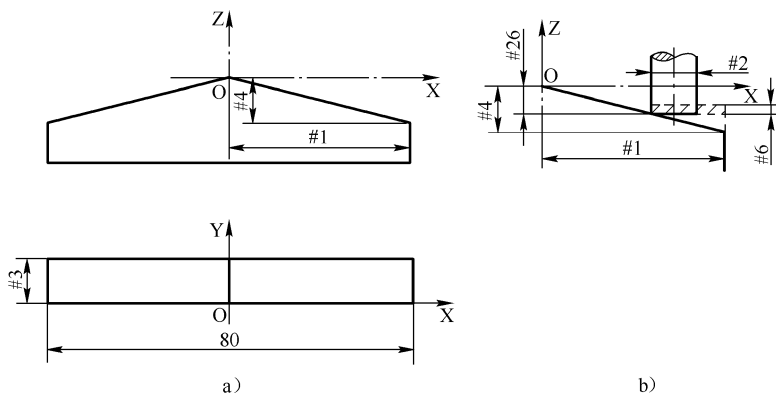


图 6-84 斜面粗加工

a) 零件图 b) 尺寸计算图

当立铣刀下深#26 时, 计算出每层切削宽度#9=#1*[1-#26/#4], 行间距为#8=0.8*#2, 每层铣削次数#10=FIX[#9/#8]上取整, 由右往内, 第一刀刀具中心 X 轴坐标为#24=#1-0.3*#2, Y 轴方向铣削斜面轮廓至#25=#3+2, 第二刀直线切削到

#24 在第一刀基础上减去一个行间距#8 的位置, Y 轴方向再铣削斜面轮廓至 #25=-2, 以后如果还有余量, 每刀#24 依次递减行间距#8, Y 轴方向前后往返铣削斜面轮廓, 直到切除该层所有粗加工余量。

完成一层铣削后, 抬刀 2mm, 快速退回到加工起始点 ([#1+#2/2], -2); 深度递增#6, 计算相应水平余量、需要铣削次数、各刀 X 轴坐标值, 再依次铣削, 完成该层加工, 直至整个右半斜面的粗加工。

最后运用镜像指令, 完成左半斜面的粗加工。

2) 变量及说明:

#1=[A]; 斜面长度

#2=[B]; 刀具直径

#3=[C]; 斜面宽度

#4=[I]; 斜面高度

#6=[K]; 深度增量

#7=[D]; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6551; 子程序号

#8=0.8*#2; 计算行间距

#11=#1+#2/2; 确定 X 轴方向加工起始坐标

G00 X#11 Y-2; 快速定位到加工起始位置

Z5; 快速下降到加工起始深度

#26=#6; 给背吃刀量变量#26 赋初始值#6

WHILE[#26LE#4] DO2; 循环 2 设定条件: 背吃刀量变量#26 不大于斜面高度#4

#9=#1*[1-#26/#4]; 计算相对于背吃刀量变量#26, X 轴方向的切削余量宽度

#10=FIX[#9/#8]; 计算 X 轴方向切削余量宽度需要切削次数

#15=0; 给 X 轴方向切削次数计数器赋初始值 0

WHILE[#15LE#10] DO1; 循环 1 设定条件: 切削次数计数器值不大于#10

#24=#1-0.3*#2-#8*#15; 计算 X 轴方向每次切削对应的刀具中心 X 轴坐标

#14=#1-#9+#2/2; 设定刀具在 X 轴方向最小极限尺寸

IF[#24LT#14] THEN#24=#14; 如果计算出的刀具中心 X 轴坐标小于其最小极限尺寸, 则限定它不小于#14

G01 X#24 F#7; 直线切削到相应刀具中心的 X 轴坐标值处

Z-#26; 直线切削到背吃刀量#26

Y[#3+2]; 斜面切削

#15=#15+1; 切削次数计数器值递增 1
IF[#15GT#10] GOTO10; 如果计数器值大于#10, 跳转到顺序号 10 程序段
#24=#1-0.3*#2-#8*#15; 计算 X 轴方向每次切削对应的刀具中心 X 轴坐标
IF[#24LT#14] THEN#24=#14; 如果计算出的刀具中心 X 轴坐标小于其最小
极限尺寸, 则限定它不小于#14
G01 X#24 F#7; 直线切削到相应刀具中心的 X 轴坐标值处
Y-2; 斜面切削
#15=#15+1; 切削次数计数器值递增 1
END1; 循环 1 结束
N10 G00 Z[2-#26]; 顺序号 10 程序段: 抬刀 2mm
X#11 Y-2; 返回加工起始点
#26=#26+#6; 背吃刀量#26 递增一个深度递增量#6
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50 处
M99; 子程序结束
%
4) 主程序如下:
%
O551;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G52 Y-5; 以 (0, -5) 为局部坐标系原点
G65 P6551 A40 B16 C10 I10 K1.5 D300; 调用宏程序 O6551, 给
相关变量赋初始值
G51.1 X0; 以 Y 轴为镜像轴建立镜像坐标系
G65 P6551 A40 B16 C10 I10 K1.5 D300; 调用宏程序 O6551, 给
相关变量赋初始值
G50.1 X0; 取消以 Y 轴为镜像轴的镜像坐标系
G52 Y0; 取消局部坐标系, 恢复 G54 坐标系
G00 G49 Z100;
X0 Y150;
M05;
M30;
%

2. 斜面精加工

1) 分析: 建立如图 6-85 所示坐标系, 采用从下而上、每层厚度为#6, 沿斜面轮廓 Y 轴方向前后往复铣削, 先加工右半斜面, 再利用镜像指令, 加工左半斜面。

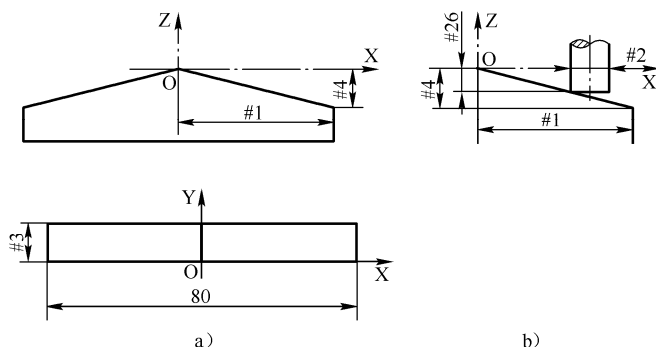


图 6-85 斜面精加工

a) 零件图 b) 尺寸计算图

开始加工时, 首先定位于起始点 ($\#1+\#2/2, -\#2/2, -\#4$) 进行沿 Y 轴正向的铣削加工至 $\#25=\#3+2$, 完成一层铣削后, 计算背吃刀量 $\#26$ 递减一个 $\#6$ 时, 相应的 X 轴坐标值 $\#24=\#2/2+\#1*\#26/\#4$, 抬刀到 $Z=-\#26$, 直线切削到 $X=\#24$, 再沿 Y 轴负向铣削加工至 $\#25=-2$, 依此类推, 直至完成整个右半斜面的精加工。

最后运用镜像指令, 完成左半斜面的精加工。

2) 变量及说明:

$\#1=[A]$; 总长度

$\#2=[B]$; 刀具直径

$\#3=[C]$; 斜面宽度

$\#4=[I]$; 斜面高度

$\#6=[K]$; 深度增量

$\#7=[D]$; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6552; 子程序号

$\#11=\#1+\#2/2$; 确定 X 轴方向加工起始坐标

G00 X $\#11$ Y $[-\#2/2]$; 快速定位到加工起始位置

Z $-\#4$; 快速下降到背吃刀量起始位置

$\#26=\#4$; 给背吃刀量变量 $\#26$ 赋初始值 $\#4$

WHILE $[\#26\text{GE}\#4]$ DO1; 循环 1 设定条件: 背吃刀量变量 $\#26$ 不小于 0

$\#24=\#2/2+\#1*\#26/\#4$; 计算相对于背吃刀量变量 $\#26$, X 轴方向的刀具中心位置

```

G01 Z-#26 F#7; 直线切削到背吃刀量#26
X#24; 直线切削到 X 轴方向刀具中心加工起始位置
Y[#3+2]; 斜面切削
#26=#26-#6; 背吃刀量#26 递减一个深度递增量#6
IF[#26LT0] GOTO10; 如果背吃刀量#26 小于 0 跳转到顺序号 10 程序段
#24=#2/2+#1*#26/#4; 计算相对于背吃刀量变量#26, X 轴方向的刀具中心位置
G01 Z-#26 F#7; 直线切削到背吃刀量#26
X#24; 直线切削到 X 轴方向刀具中心加工起始位置
Y-2; 斜面切削
#26=#26-#6; 背吃刀量#26 递减一个深度递增量#6
END1; 循环 1 结束
N10 G00 Z50; 顺序号 10 程序段: 快速上升到 Z=50 处
M99; 子程序结束
%
```

4) 主程序如下:

```

%
O652;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G52 Y-5; 以 (0, -5) 为局部坐标系原点
G65 P6552 A40 B16 C10 I10 K0.1 D300; 调用宏程序 O6552, 给
                                         相关变量赋初始值
G51.1 X0; 以 Y 轴为镜像轴建立镜像坐标系
G65 P6552 A40 B16 C10 I10 K0.1 D300; 调用宏程序 O6552, 给
                                         相关变量赋初始值
G50.1 X0; 取消以 Y 轴为镜像轴的镜像坐标系
G52 Y0; 取消局部坐标系, 恢复 G54 坐标系
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

3. 倒圆斜面精加工

1) 分析: 建立如图 6-86 所示坐标系, 采用从下而上、先进行圆角的铣削加

工, 初始角度为 0° , 以后角度依次以角度增量#6 递增, 计算出相应的 X、Z 轴坐标#24、#26, Y 轴向前后往复铣削, 完成圆角加工。然后进行斜面的加工, 方法参照图 6-85。

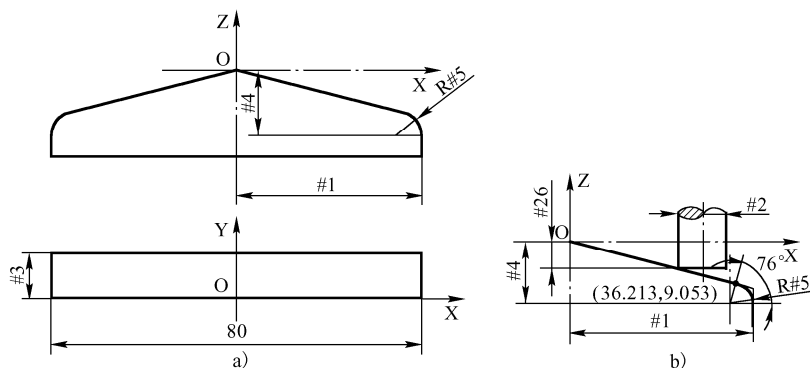


图 6-86 倒圆斜面精加工

a) 零件图 b) 尺寸计算图

最后运用镜像指令, 完成左半斜面的精加工。

2) 变量及说明:

#1=[A]; 斜面长度

#2=[B]; 刀具直径

#3=[C]; 厚度

#4=[I]; 高度

#5=[J]; 倒圆半径

#6=[K]; 深度增量

#7=[D]; 进给量, 单位为 mm/min

#8=[E]; 角度增量

3) 宏程序如下:

%

O6553; 子程序号

G00 X[#1+#2/2] Y0; 快速定位到加工起始位置

Z-#4; 快速下降到背吃刀量起始位置

#20=76; 将切点对应的角度值赋给变量#20

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE#20] DO1; 循环 1 设定条件: 角度变量#8 不大于#20

#24=#1-#5*[1-COS[#8]]+#2/2; 计算对应于角度变量#8 刀具中心的 X 轴坐标

#26=#4-#5*SIN[#8]; 计算对应于角度变量#8 刀具中心的 Z 轴坐标

```
G01 X#24 Z-#26 F#7; 直线切削到加工起始位置
Y[#3+2]; 倒圆切削
#8=#8+#6; 角度变量#8 递增一个角度递增量#6
IF[#8GT#20] GOTO10; 如果角度变量#8 大于#20 跳转到顺序号 10 程序段
#24=#1-#5*[1-COS[#8]]+#2/2; 计算对应于角度变量#8 刀具中心的 X 轴坐标
#26=#4-#5*SIN[#8]; 计算对应于角度变量#8 刀具中心的 Z 轴坐标
G01 X#24 Z-#26 F#7; 直线切削到加工起始位置
Y-2; 倒圆切削
#8=#8+#6; 角度变量#8 递增一个角度递增量#6
END1; 循环 1 结束
N10 G01 Y-2; 顺序号 10 程序段: 轮廓切削
#10=36.213; 赋相关变量初始值
#9=9.053; 赋相关变量初始值
#6=0.1; 赋相关变量初始值
#26=#9; 给背吃刀量变量#26 赋初始值#9
WHILE[#26GE0] DO2; 循环 2 设定条件: 深度变量#26 不小于 0
#24=#10*#26/#9+#2/2; 计算相对于深度变量#26 刀具中心 X 轴坐标
G01 X#24 Z-#26; 直线切削到加工起始位置
Y[#3+2]; 斜面切削
#26=#26-#6; 深度变量#26 递减一个深度递增量#6
IF[#26LT0] GOTO20; 如果深度变量#26 小于 0 跳转到顺序号 20 的程序段
#24=#10*#26/#9+#2/2; 计算相对于深度变量#26 刀具中心 X 轴坐标
G01 X#24 Z-#26; 直线切削到加工起始位置
Y-2; 斜面切削
#26=#26-#6; 深度变量#26 递减一个深度递增量#6
END2; 循环 2 结束
N20 G00 Z50; 顺序号 20 程序段: 快速抬刀至 Z=50 处
M99; 子程序结束
%
4) 主程序如下:
%
O653;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
```

G52 Y-5; 以 (0, -5) 为局部坐标系原点

G65 P6553 A40 B16 C10 I14 J5 K2 D300; 调用宏程序 O6553,
给相关变量赋初始值

G51.1 X0; 以 Y 轴为镜像轴建立镜像坐标系

G65 P6553 A40 B16 C10 I14 J5 K5 D300; 调用宏程序 O6553,
给相关变量赋初始值

G50.1 X0; 取消以 Y 轴为镜像轴的镜像坐标系

G52 Y0; 取消局部坐标系, 恢复 G54 坐标系

G00 G49; Z100;

X0 Y150;

M05;

M30;

%

6.5.6 实例六圆柱面加工

1. 外凸圆柱面

(1) 粗加工

如图 6-87 所示, 外凸圆柱面进行粗加工。其中圆柱体半径为 #1, 厚度为 #3, 立铣刀直径 #2。

1) 分析: 建立如图 6-87 所示坐标系, 采用从上而下、每层厚度为 #6、由右以行间距 #8=0.8*#2 向中心逐次铣削, 先加工右半圆柱, 再利用镜像指令, 加工左半圆柱。基本进给路线: A→B→C→D→E 依次递进, 厚度方向前后都多出 2mm。

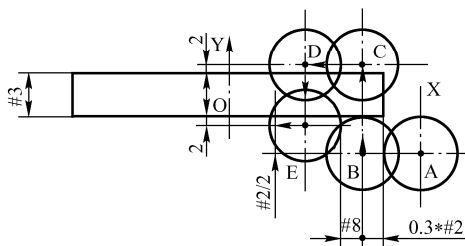
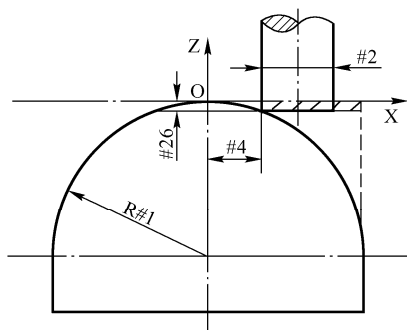


图 6-87 外凸圆柱面粗加工

当立铣刀下深 #26 时, $\#4 = \sqrt{\#1^2 - (\#1 - \#26)^2}$, 需要铣削的余量就是 #1-#4,

行间距为 #8=0.8*#2, 每层铣削次数 #10=(#1-#4)/#8 上取整, 由右往内, 第一刀刀具中心 X 轴坐标为 #1-0.3*#2, 第二刀及以后递减行间距 #8, 直到切除余量 #1-#4。

完成一层铣削后, 抬刀 2mm, 快速退回到加工起始点 ([#1+#2/2], -2); 深度递增 #6, 计算相应水平余量、需要铣削次数、各刀 X 轴坐标值, 再依次铣削, 完成该层加工, 直至整个右半圆柱面的粗加工。

最后运用镜像指令, 完成左半圆柱面的粗加工。

2) 变量及说明:

#1=[A]; 圆柱面半径

#2=[B]; 刀具直径

#3=[C]; 圆柱面宽度

#6=[K]; 深度增量

#7=[D]; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6561; 子程序号

#8=0.8*#2; 计算行间距

G00 X[#1+#2/2] Y-[#2/2]; 快速定位到加工准备位置

Z5; 快速下降到 Z=5

#26=#6; 给背吃刀量赋初始值一个背吃刀量增量#6

WHILE[#26LE#1] DO2; 循环 2 设定条件: 背吃刀量#26 不大于圆柱半径

#100=[#1-#26]*[#1-#26]; 中间值寄存

#4=SQRT[#1*#1-#100]; 计算任意深度对应的弦长半值

#14=#4+#2/2; 刀具最小 X 轴极限坐标

G01 Z-#26 F#7; 直线切削到相应深度

IF[#1-#4LT#2] GOTO10; 如果余量小于刀具直径, 跳转到顺序号 10

#10=FIX[[(#1-#4)/#8]; 计算水平切削次数

#15=0; 给水平切削次数计数器值赋 0

WHILE[#15LE#10] DO1; 循环 1 设定条件: 计数器值不大于计算次数

#24=#1-0.3*#2-#8*#15; 计算相应轮次 X 轴向坐标值

IF[#24LT#14] THEN#24=#14; 限定 X 轴向最小极限尺寸

G01 X#24; 快速定位到本轮次 X 轴向坐标值处

Y[#3+2]; 直线切削圆柱轮廓

#15=#15+1; 次数计数器值累加 1

IF[#15GT#10] GOTO20; 如果次数计数器值大于计算水平切削次数跳转到顺序号 20

#24=#1-0.3*#2-#8*#15; 计算相应轮次 X 轴向坐标值

IF[#24LT#14] THEN#24=#14; 限定 X 轴向最小极限尺寸

G01 X#24; 直线切削到本轮次 X 轴向坐标值处

Y-2; 直线切削圆柱轮廓

#15=#15+1; 次数计数器值累加 1

END1; 循环 1 结束

```
GOTO20; 跳转到顺序号 20
N10 G01 X#14; 快速左移到刀具最小 X 轴极限坐标
Y[#3+2]; 直线切削轮廓
N20 G00 Z[2-#26]; 抬刀 2mm
X[#1+#2/2] Y-2; 返回到加工起始位置
#26=#26+#6; 背吃刀量递增一个背吃刀量增量#6
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50
M99; 子程序结束
%
```

4) 主程序如下:

```
%
O561;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G52 Y-5
G65 P6561 A40 B16 C10 K1.5 D300; 调用宏程序 O6561, 给相关变
量赋初始值
G51.1 X0; 建立以 Y 轴为镜像轴的镜像坐标系
G65 P6561 A40 B16 C10 K1.5 D300; 调用宏程序 O6561, 给相关变
量赋初始值
G50.1 X0; 取消以 Y 轴为镜像轴的镜像坐标系
G52 Y0;
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

(2) 精加工

如图 6-88 所示, 外凸圆柱面进行精加工。其中圆柱体半径为#1, 厚度为#3, 立铣刀直径#2。

1) 分析: 建立如图 6-88 所示坐标系, 采用从下而上、每次角度递增#6、由右端向中心上方逐次铣削, 先右半圆柱, 再利用镜像指令, 加工左半圆柱。厚度方向前后都多出 2mm。

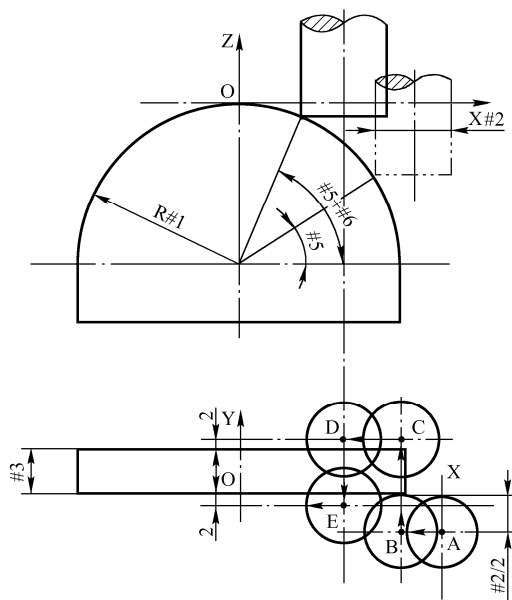


图 6-88 外凸圆柱面精加工

刚开始从最右端，此时角度为 $\#5=0^\circ$ ，立铣刀中心顶点坐标为 $X=\#24=\#1*\cos[\#5]+\#2/2$ ， $Z=\#26=-\#1*[1-\sin[\#5]]$ ；第二刀角度为 $\#5=\#6$ ，立铣刀中心顶点坐标为 $X=\#24=\#1*\cos[\#6]+\#2/2$ ， $Z=\#26=-\#1*[1-\sin[\#6]]$ ；第三刀及以后角度 $\#5=\#5+\#6$ ；直至角度为 90° ，完成右半圆柱面的精加工。

最后运用镜像指令，完成左半圆柱面的精加工。

2) 变量及说明：

$\#1=[A]$ ；圆柱面半径

$\#2=[B]$ ；刀具直径

$\#3=[C]$ ；圆柱面宽度

$\#6=[K]$ ；角度增量

$\#7=[D]$ ；进给量，单位为 mm/min

3) 宏程序如下：

%

O6562；子程序号

G00 X[#1+#2/2] Y-[#2/2]；快速定位到加工准备位置

Z[2-#1]；快速下降到距背吃刀量 2mm 处

$\#5=0$ ；切削夹角赋初始值 0

WHILE[#5LE90] DO2；循环 2 设定条件：夹角不大于 90°

```
#24=#1*COS[#5]+#2/2; 计算相应 X 轴坐标值
#26=#1*[SIN[#5]-1]; 计算相应 Z 轴坐标值
G01 Z#26 F#7; 直线切削至背吃刀量
X#24; 直线切削至切削起始位置
Y[#3+2]; 轮廓切削
#5=#5+#6; 夹角递增一个角度增量#6
IF[#5GT90] GOTO20; 如果夹角大于 90°, 跳转到顺序号 20
#24=#1*COS[#5]+#2/2; 计算相应 X 轴坐标值
#26=#1*[SIN[#5]-1]; 计算相应 Z 轴坐标值
G01 Z#26 F#7; 直线切削至背吃刀量
X#24; 直线切削至切削起始位置
Y-2; 轮廓切削
#5=#5+#6; 夹角递增一个角度增量#6
END2; 循环 2 结束
N20 G01 X0; 顺序号 20 程序段直线切削到圆柱面顶部
Y[#3+2]; 轮廓切削
Y-2; 轮廓切削往复
G00 Z50; 快速抬刀到 Z=50
M99; 子程序结束
%
```

4) 主程序如下:

```
%
O562;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G52 Y-5; 建立局部坐标系, 以 (0, -5, 0) 为局部原点
G65 P6562 A40 B16 C10 K5 D300; 调用宏程序 O6562, 给相关变量
    赋初始值
G51.1 X0; 建立以 Y 轴为镜像轴的镜像坐标系
G65 P6562 A40 B16 C10 K5 D300; 调用宏程序 O6562, 给相关变量
    赋初始值
G50.1 X0; 取消以 Y 轴为镜像轴的镜像坐标系
G52 Y0; 取消局部坐标系, 还原 G54 坐标系
G00 G49 Z100;
```

X0 Y150;

M05;

M30;

%

2. 倒圆角外凸圆柱面

(1) 粗加工

如图 6-89 所示, 倒圆角外凸圆柱面进行粗加工。其中圆柱体半径为#1, 厚度为#3, 立铣刀直径为#2。

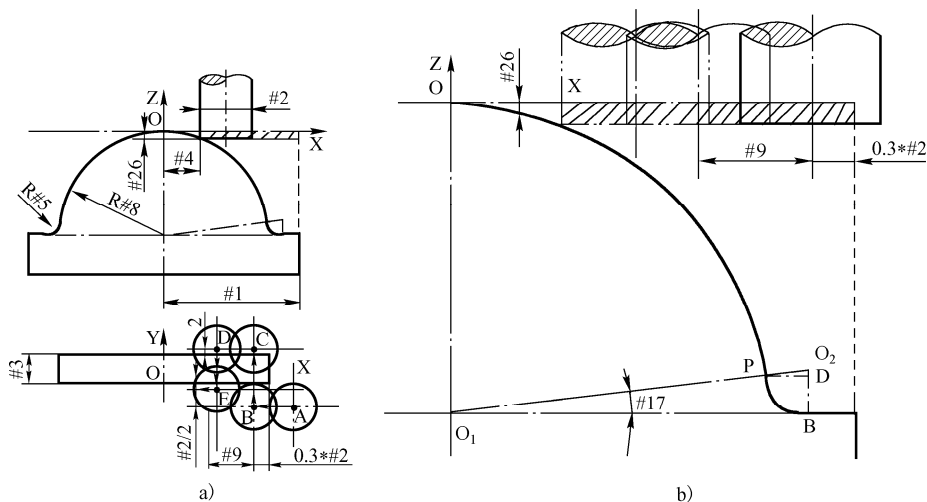


图 6-89 倒圆角外凸圆柱面粗加工

a) 加工示意图 b) 局部放大图

1) 分析: 由图 6-89b 可知, 按照图 6-87 加工思路, 由上而下, 从 O 点到 P 点轮廓粗加工程序编制与 O6561 相同, 而从 P 点到 B 点轮廓属于凹圆柱面粗加工; 首先需要算出 P 点坐标, 在 $\triangle O_1BO_2$ 中, $O_1O_2=\#20=\#8+\#5$, $O_2B=\#5$, $O_1B=\#21=\text{SQRT}[\#20*\#20-\#5*\#5]$, $\angle O_2O_1B=\#17=\text{ATAN}[\#5]/[\#21]$ 。

则 P 点坐标:

$$X=\#13=\#8*\text{COS}[\#17]$$

$$Z=\#14=\#8*[\text{SIN}[\#17]-1]$$

整个粗加工都是以背吃刀量#26 为变量, #6 为递增量。第一阶段背吃刀量#26 从 0 到#14, 相关计算、编程参照图 6-87; 第二阶段背吃刀量#26 从#14 到#8, 刀具中心顶点每层左极限位置计算如下:

$$\#22=\#5*\text{SIN}[\#17]+\#26+\#14$$

$$\#4=\text{SQRT}[\#5*\#6-\#22*\#22]$$

$X=\#21+\#2/2-\#4$

每层加工余量:

$\#31=\#1/2-\#13$

$\#10=\text{FIX}[\#31/\#9]$

其余计算、编程方法与图 6-87 相同。

2) 变量说明:

$\#1=[A]$; 圆柱面半径

$\#2=[B]$; 刀具直径

$\#3=[C]$; 圆柱面宽度

$\#6=[K]$; 深度增量

$\#7=[D]$; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6563; 子程序号

$\#9=0.8*\#2$; 行间距

G00 X $[(\#1+\#2)/2]$ Y $[-\#2/2]$; 快速定位到准备位置

Z5; 快速下降到 Z=5 处

$\#20=\#8+\#5$; 计算中心距 O_1O_2

$\#21=\text{SQRT}[\#20*\#20-\#5*\#5]$; 计算 O_1B 长度

$\#17=\text{ATAN}[\#5]/[\#21]$; 计算 $\angle O_2O_1B$ 度数

$\#13=\#8*\text{COS}[\#17]$; 计算切点 P 的 X 坐标

$\#14=\#8*[\text{SIN}[\#17]-1]$; 计算切点 P 的 Z 坐标

$\#26=\#6$; 给背吃刀量变量 $\#26$ 赋初始值 $\#6$

WHILE $[\#26\text{LE}-\#14]$ DO2; 循环 2 设定条件: 背吃刀量变量 $\#26$ 不大于切点 P 深度 $\#14$, 即限定只加工 R $\#8$ 圆弧

G01 Z $-\#26$ F $\#7$; 直线切削到每层深度 $\#26$

$\#100=[\#8-\#26]*[\#8-\#26]$; 中间值寄存在变量 $\#100$

$\#4=\text{SQRT}[\#8*\#8-\#100]$; 计算 R $\#8$ 圆弧圆心到每层切削面距离 $\#4$

$\#30=\#4+\#2/2$; 计算刀具中心顶点左极限位置

$\#31=\#1/2-\#4$; 计算每层切削余量

IF $[\#31\text{LT}\#2]$ GOTO10; 如果每层切削余量小于刀具直径, 跳转到顺序号 10 程序段, 一次完成该层切削

$\#10=\text{FIX}[\#31/\#9]$; 计算每层需要切削次数

$\#15=0$; 给切削次数计数器赋初始值 0

WHILE $[\#15\text{LE}\#10]$ DO1; 循环 1 设定条件: 计数器值不大于 $\#10$

#24=#1/2-0.3*#2-#9*#15; 计算刀具中心 X 轴坐标
IF[#24LT#30] THEN#24=#30; 如果刀具中心 X 轴坐标小于其左极限尺寸,
则限定它最小为其左极限尺寸
G01 X#24; 直线切削到计算的 X 轴坐标位置
Y[#3+2]; 轮廓切削
#15=#15+1; 计数器累加 1
IF[#15GT#10] GOTO20; 如果计数器值大于#10, 跳转到顺序号 20 程序段
#24=#1/2-0.3*#2-#9*#15; 计算刀具中心 X 轴坐标
IF[#24LT#30] THEN#24=#30; 如果刀具中心 X 轴坐标小于其左极限尺寸,
则限定它最小为其左极限尺寸
G01 X#24; 直线切削到计算的 X 轴坐标位置
Y-2; 轮廓切削
#15=#15+1; 计数器累加 1
END1; 循环 1 结束
GOTO20; 跳转到顺序号 20 程序段
N10 G01 X#30; 顺序号 10 程序段: 直线切削到刀具中心左极限位置
Y[#3+2]; 轮廓切削
N20 G00 Z[2-#26]; 顺序号 20 程序段: 抬刀 2mm
X[#1/2+#2/2] Y[-#2/2]; 快速位移到下层切削准备位置
#26=#26+#6; 背吃刀量递增#6
END2; 循环 2 结束
#26=-#14; 给背吃刀量赋初始值 P 点深度
WHILE[#26LE#8] DO2; 循环 2 设定条件: 背吃刀量变量#26 不大于总深度
#8, 即限定只加工倒圆部分
#22=#5*SIN[#17]+#26+#14; 计算倒圆部分圆心到每层水平切削面的距离
#4=SQRT[#5*#6-#22*#22]; 计算倒圆部分每层水平切削面宽度半值
#30=#21-#4+#2/2; 计算刀具中心顶点左极限位置
#31=#1/2-#13; 计算每层切削余量
#10=FIX[#31/#9]; 计算每层需要切削次数
G01 Z-#26 F#7; 直线切削到背吃刀量#26
#16=0; 给切削次数计数器赋初始值 0
WHILE[#16LE#10] DO1; 循环 1 设定条件: 计数器值不大于#10
#24=#1/2-0.3*#2-#9*#16; 计算刀具中心 X 轴坐标
IF[#24LT#30] THEN#24=#30; 如果刀具中心 X 轴坐标小于其左极限尺寸,
则限定它最小为其左极限尺寸

G01 X#24; 直线切削到计算的 X 轴坐标位置
Y[#3+2]; 轮廓加工
#16=#16+1; 计数器累加 1
IF[#16GT#10] GOTO30; 如果计数器值大于#10, 跳转到顺序号 30 程序段
#24=#1-0.3*#2-#9*#16; 计算刀具中心 X 轴坐标
IF[#24LT#30] THEN#24=#30; 如果刀具中心 X 轴坐标小于其左极限尺寸,
则限定它最小为其左极限尺寸

G01 X#24; 直线切削到计算的 X 轴坐标位置
Y-2; 轮廓加工
#16=#16+1; 计数器累加 1
END1; 循环 1 结束
N30 G00 Z[2-#26]; 顺序号 30 程序段: 抬刀 2mm
X[#1/2+#2/2] Y[-#2/2]; 快速位移到下层切削准备位置
#26=#26+#6; 背吃刀量递增#6
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50 处
M99; 子程序结束
%

4) 主程序如下:

%
O663;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G52 Y-5; 建立局部坐标系, 以 (0, -5, 0) 为局部原点
G65 P6563 A80 B16 C10 J5 K1.5 D300 E30; 调用宏程序 O6563,
给相关变量赋初始值
G51.1 X0; 建立以 Y 轴为镜像轴的镜像坐标系
G65 P6563 A80 B16 C10 J5 K1.5 D300 E30; 调用宏程序 O6563,
给相关变量赋初始值
G50.1 X0; 取消以 Y 轴为镜像轴的镜像坐标系
G52 Y0; 取消局部坐标系, 还原 G54 坐标系
G00 G49 Z100;
X0 Y150;
M05;

M30;

%

(2) 精加工

如图 6-90 所示, 倒圆角外凸圆柱面进行精加工。其中圆柱体半径为#1, 厚度为#3, 立铣刀直径为#2。

1) 分析: 由图 6-90 可知, 由下而上, 从右往左, 首先精加工倒圆部分, 然后精加工圆弧 $R\#8$ 部分。以角度为变量, #6 角增量值, 两段角度范围分别为:

第一段:

由前面例子可知 $\#9 = \text{ATAN}[\#5]/[\#21]$

(这里#9=前面的#17)

倒圆部分角度极限值: $\#16 = 90 - \#9$

角度范围: $0^\circ \sim \#16$

第二段:

角度范围: $\#9 \sim 90^\circ$

2) 变量及说明:

#1=[A]; 底面长度

#2=[B]; 刀具直径

#3=[C]; 圆柱面宽度

#5=[J]; 倒圆半径

#6=[K]; 角度增量

#7=[D]; 进给量, 单位为 mm/min

#8=[E]; 圆柱面半径

3) 宏程序如下:

%

O6564; 子程序号

G00 X[#1/2+#2/2] Y[-#2/2]; 快速定位到加工准备位置

Z[2-#8]; 快速下降到距背吃刀量 2mm 处

#20=#8+#5; 计算中心距 O_1O_2

#21= $\text{SQRT}[\#20^2 - \#5^2]$; 计算 O_1B 长度

#9= $\text{ATAN}[\#5]/[\#21]$; 计算 $\angle O_2O_1B$ 度数

#13=#8*COS[#9]; 计算切点 P 的 X 坐标

#14=#8*[SIN[#9]-1]; 计算切点 P 的 Z 坐标

#15=0; 给角度变量#15 赋初始值 0

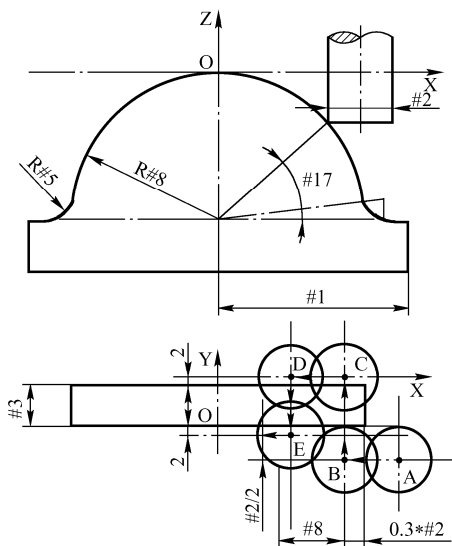


图 6-90 倒圆角外凸圆柱面精加工

```
#16=90-#9; 计算角度极限值
WHILE[#15LE#16] DO1; 循环 1 设定条件: 角度变量#15 不大于#16
#26=#8*[1-SIN[#17]]; 计算下一轮次 Z 坐标
#24=#8*COS[#17]+#2/2; 计算下一轮次 X 坐标
G01 Z-#26 F#7; 直线切削到背吃刀量
X#24; 直线切削到计算的 X 轴坐标位置
Y[#3+2]; 轮廓加工
#15=#15+#6; 角度变量#15 递增#6
IF[#15GT#16] GOTO10; 如果角度变量#15 大于#16, 跳转到顺序号 10 程序段
#26=#8*[1-SIN[#17]]; 计算下一轮次 Z 坐标
#24=#8*COS[#17]+#2/2; 计算下一轮次 X 坐标
G01 Z-#26 F#7; 直线切削到背吃刀量
X#24; 直线切削到计算的 X 轴坐标位置
Y-2; 轮廓加工
#15=#15+#6; 角度变量#15 递增#6
END1; 循环 1 结束
N10 G01 Y-2; 顺序号 10 程序段: 直线切削轮廓
#17=#9; 给角度变量#17 赋初始值#9
WHILE[#17LE90] DO2; 循环 2 设定条件: 角度变量#17 不大于 90°
#26=#8*[1-SIN[#17]]; 计算下一轮次 Z 坐标
#24=#8*COS[#17]+#2/2; 计算下一轮次 X 坐标
G01 Z-#26 F#7; 直线切削到背吃刀量
X#24; 直线切削到计算的 X 轴坐标位置
Y[#3+2]; 轮廓加工
#17=#17+#6; 角度变量递增#6
IF[#17GT90] GOTO20; 如果角度变量大于 90°, 跳转到顺序号 20 程序段
#26=#8*[1-SIN[#17]]; 计算下一轮次 Z 坐标
#24=#8*COS[#17]+#2/2; 计算下一轮次 X 坐标
G01 Z-#26 F#7; 直线切削到背吃刀量
X#24; 直线切削到计算的 X 轴坐标位置
Y-2; 轮廓加工
#17=#17+#6; 角度变量递增#6
END2; 循环 2 结束
N20 G01 Z0; 顺序号 20 程序段: 直线切削到 Z0
X0; 切削到 X0, 准备弧顶切削
```


Y[#3+2]; 弧顶切削

Y-2; 弧顶切削

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

O564;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G52 Y-5; 建立局部坐标系, 以 (0, -5, 0) 为局部原点

G65 P6564 A80 B16 C10 J5 K5 D300 E30; 调用宏程序 O6564,
给相关变量赋初始值

G51.1 X0; 建立以 Y 轴为镜像轴的镜像坐标系

G65 P6564 A80 B16 C10 J5 K5 D300 E30; 调用宏程序 O6564,
给相关变量赋初始值

G50.1 X0; 取消以 Y 轴为镜像轴的镜像坐标系

G52 Y0; 取消局部坐标系, 还原 G54 坐标系

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

3. 内凹圆柱面

(1) 粗加工

如图 6-91 所示, 对内凹圆柱面进行粗加工。其中圆柱体半径为 #1, 厚度为 #3, 立铣刀直径为 #2。

1) 分析: 由于立铣刀有一定直径, 内凹圆柱底部有部分材料将无法加工到, 所以需要计算出相关尺寸, 从而确定立铣刀能加工到的深度。

从图 6-91 可知, 此深度即是: $\#4 = \text{SQRT}[\#1 * \#1 - [\#2/2] * [\#2/2]]$;

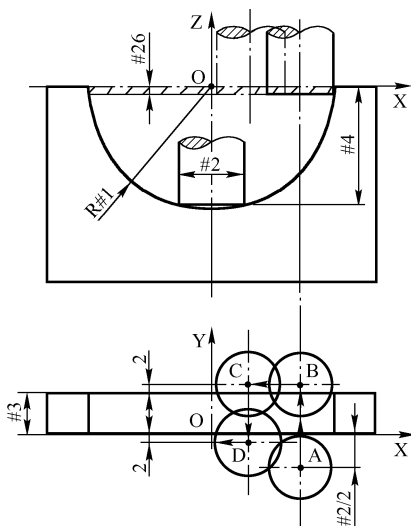


图 6-91 内凹圆柱面粗加工

每次背吃刀量为#6，需铣削次数： $\#4/\#6$ ；

铣削深度：#26 从#6 开始，每层递增#6，直到#4；

宽度： $\#10=\text{SQRT}[\#1*\#1-\#26*\#26]*2$ ；

行间距： $\#8=0.8*\#2$ ；

每层需铣削次数： $\#11=\text{FIX}[\#10/\#8]$ ；

刀具中心从最右端极限位置 $\#24=[\#10-\#2]/2-\#8*\#15$ 开始，依次向左到刀具中心最左端极限位置 $[\#2-\#10]/2$ 。

2) 变量及说明：

#1=[A]；内凹圆柱面半径

#2=[B]；刀具直径

#3=[C]；厚度

#6=[K]；深度增量

#7=[D]；进给量，单位为 mm/min

3) 宏程序如下：

%

O6565；子程序号

G00 X[#1-#2/2] Y[-#2/2]；快速定位到加工起始位置

Z5；快速下降到 Z=5 深度起始高度

#8=0.8*#2；设定行间距

#20=[#2/2]*[#2/2]；中间值寄存

#4=SQRT[#1*#1-#20]；计算立铣刀所能加工深度

#26=#6；给背吃刀量变量#26 赋初始值：#6

WHILE[#26LE#4] DO2；循环 2 设定条件：背吃刀量#26 不大于所能加工深度#4

#10=SQRT[#1*#1-#26*#26]*2；计算切削宽度

#20=[#2-#10]/2；刀具中心左右极限位置坐标

#11=FIX[#10/#8]；计算每层需要切削次数

#15=0；给切削次数计数器赋初始值 0

WHILE[#15LE#11] DO1；循环 1 设定条件：计数器值#15 不大于每层需要切削次数#11

#24=[#10-#2]/2-#8*#15；计算每层切削开始 X 轴方向位置

IF[#24LT#20] THEN#24=#20；限定刀具中心最左端位置

G01 X#24 F#7；水平左右位置定位

Z-#26；下深到背吃刀量位置

Y[#3+2]；直线切削轮廓

```
#15=#15+1; 计数器累加 1
IF[#15GT#11] GOTO10; 如果计数器大于#11, 跳转到顺序号 10
#24=[#10-#2]/2-#8*#15; 计算每层切削开始 X 轴方向位置
IF[#24LT#20] THEN#24=#20; 限定刀具中心最左端位置
G01 X#24 F#7; 水平左右位置定位
Y-2; 直线切削轮廓
#15=#15+1; 计数器累加 1
END1; 循环 1 结束
N10 G00 Z[2-#26]; 抬刀 2mm
X[#10-#2]/2 Y-2; 返回加工起始位置
#26=#26+#6; 背吃刀量递增#6
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50
M99; 子程序结束
%
```

4) 主程序如下:

```
%
O565;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G52 Y-5; 建立局部坐标系, 以 (0, -5, 0) 为原点
G65 P6565 A35 B16 C10 K1.5 D300; 调用宏程序 O6565, 给相关变
量赋初始值
G52 Y0; 取消局部坐标系, 恢复 G54 坐标系
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

(2) 精加工

如图 6-92 所示, 倒圆角内凹圆柱面进行精加工。其中圆柱体半径为#1, 厚度为#3, 球头铣刀半径为#2。

1) 分析: 从内凹圆柱面中心底部开始加工, 先加工右半部分, 再利用镜像指令加工左半部分; 以球头铣刀球心与圆柱轴线连线和垂直面间的夹角#5 为变量,

该变量从 0° 开始一直到 90° ，每次递增 $\#6$ ，依次计算相应角度时刀具球心 X、Z 轴坐标，进行轮廓加工即可将内凹圆柱面精加工。

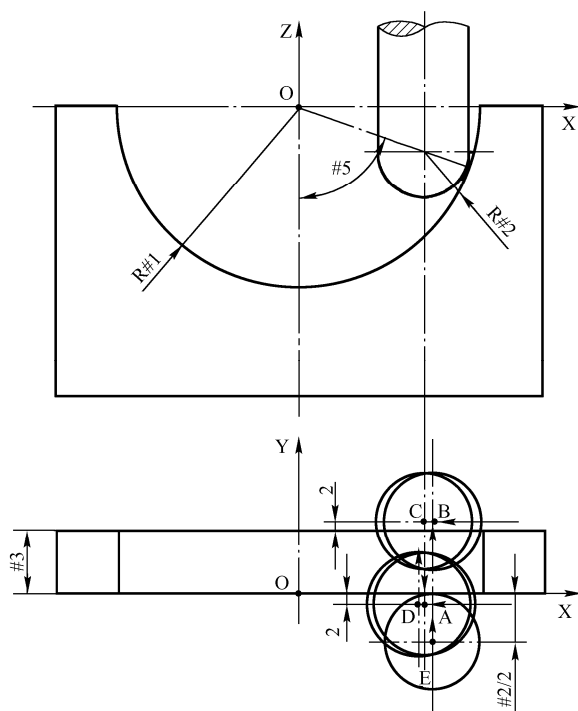


图 6-92 内凹圆柱面精加工

2) 变量及说明:

$\#1=[A]$; 内凹圆柱面半径

$\#2=[B]$; 球头铣刀半径

$\#3=[C]$; 厚度

$\#6=[K]$; 角度增量

$\#7=[D]$; 进给量, 单位为 mm/min

3) 宏程序如下:

%

O6566; 子程序号

G00 X0 Y[-#2/2]; 快速定位到加工准备位置

Z[-#1-#2-2]; 快速下降到距离开始背吃刀量 2mm 处

$\#5=0$; 给角度变量 $\#5$ 赋初始值 0

WHILE[#5LE90] DO1; 循环 1 设定条件: 角度变量 $\#5$ 不大于 90°

$\#24=[\#1-\#2]*\text{SIN}[\#5]$; 计算 X 轴坐标

```
#26=[#1-#2]*COS[#5]; 计算 X 轴坐标
G01 X#24 Z-#26 F#7; 直线切削到切削开始点
Y[#3+2]; 轮廓铣削
#5=#5+#6; 角度递增#6
IF[#5GT90] GOTO10; 如果角度#5 大于 90°, 跳转到顺序号 10 程序段
#24=[#1-#2]*SIN[#5]; 计算 X 轴坐标
#26=[#1-#2]*COS[#5]; 计算 X 轴坐标
G01 X#24 Z-#26; 直线切削到切削开始点
Y[#3+2]; 轮廓铣削
#5=#5+#6; 角度递增#6
END1; 循环 1 结束
N10 G00 Z50; 顺序号 10 程序段: 快速上升到 Z=50
M99; 子程序结束
%
```

4) 主程序如下:

```
%
O566;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G52 Y-5; 建立局部坐标系, 以 (0, -5, 0) 为原点
G65 P6566 A35 B8 C10 K5 D300; 调用宏程序 O6566, 给相关变量
    赋初始值
G51.1 X0; 建立以 Y 轴为镜像轴的镜像坐标系
G65 P6566 A35 B8 C10 K5 D300; 调用宏程序 O6566, 给相关变量
    赋初始值
G50.1 X0; 取消以 Y 轴为镜像轴的镜像坐标系
G52 Y0; 取消局部坐标系, 恢复 G54 坐标系
G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

4. 倒圆角内凹圆柱面精加工

如图 6-93 所示, 对倒圆角内凹圆柱面进行精加工。其中圆柱体半径为#1, 厚

度为#3，球头铣刀半径为#2，倒圆半径为#5。

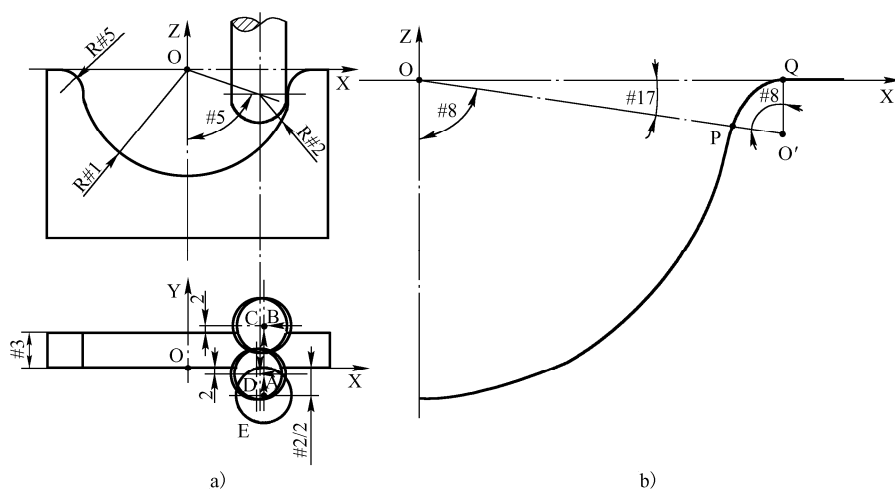


图 6-93 倒圆角内凹圆柱面精加工

a) 示意图 b) 局部放大图

1) 分析：从内凹圆柱面中心底部开始加工，先加工右半部分，再利用镜像指令加工左半部分。以球头铣刀球心与圆柱轴线连线和垂直面间的夹角为#13，以该角度作为变量，该变量从 0° 开始一直到倒圆与圆柱切线处角度#8，倒圆处角度从#8 逐步递减到 0° ，角度每次递减都是#6，依次计算相应角度时刀具球心 X、Z 轴坐标，进行轮廓加工即可将倒圆内凹圆柱面进行精加工。

从图 6-93b 中可知，圆柱与倒圆圆心以及切点 Q 组成直角 $\triangle OO'Q$ ，其中：

$OO' = \text{两半径之和} = \#1 + \#5$ ， $O'Q = \#5$ ， $OQ = \sqrt{OO'^2 - O'Q^2} = \#30 = \text{SQRT}[(\#1 + \#5) * (\#1 + \#5) - \#5 * \#5]$ ， $\angle O'OQ = \#17 = \text{ATAN}[\#5]/[\#30]$ ， $\angle OO'Q = \#8 = 90^\circ - \#17$ 。

2) 变量及说明：

#1=[A]；内凹圆柱面半径

#2=[B]；刀具直径

#3=[C]；厚度

#5=[J]；倒圆半径

#6=[K]；深度增量

#7=[D]；进给量，单位为 mm/min

3) 宏程序如下：

%

O6567；子程序号

```
G00 X0 Y[-#2/2]; 快速定位到加工准备位置
Z-[#1-#2-2]; 快速下降到距离内凹圆柱中心底面 2mm 处
#30=SQRT[(#1+#5)*[#1+#5]-#5*#5]; 计算 OQ 长度
#17=ATAN[#5]/[#30]; 计算  $\angle O'OQ$  角度
#8=90-#17; 计算  $\angle OO'Q$  角度
#13=0; 给角度变量赋初始值 0
WHILE[#13LE#8] DO1; 循环 1 设定条件: 角度变量#13 不大于#8
#24=[#1-#2]*SIN[#13]; 计算刀具球心 X 轴坐标
#26=[#1-#2]*COS[#13]; 计算刀具球心 Z 轴坐标
G01 X#24 Z-#26 F#7; 直线切削到加工轮廓起始点
Y[#3+2]; 轮廓加工
#13=#13+#6; 角度变量递增一个#6 值
IF[#13GT#8] GOTO10; 如果角度变量大于#8, 跳转到序号 10 程序段
#24=[#1-#2]*SIN[#13]; 计算刀具球心 X 轴坐标
#26=[#1-#2]*COS[#13]; 计算刀具球心 Z 轴坐标
G01 X#24 Z-#26 F#7; 直线切削到加工轮廓起始点
Y-2; 轮廓加工
#13=#13+#6; 角度变量递增一个#6 值
END1; 循环 1 结束
N10 G01 Y-2; 序号 10 程序段: 轮廓切削
#15=#8; 给角度变量#15 赋初始值#8
WHILE[#15GE0] DO2; 循环 2 设定条件: 角度变量#15 不小于 0
#24=#30-[#5+#2]*SIN[#15]; 计算刀具球心 X 轴坐标
#26=[#5+#2]*[1-COS[#15]]-#2; 计算刀具球心 Z 轴坐标
G01 X#24 Z-#26 F#7; 直线切削到加工轮廓起始点
Y[#3+2]; 轮廓加工
#15=#16-#6; 角度变量#15 递减一个#6
IF[#15LT0] GOTO20; 如果角度变量小于 0, 跳转到序号 20 程序段
#24=#30-[#5+#2]*SIN[#15]; 计算刀具球心 X 轴坐标
#26=[#5+#2]*[1-COS[#15]]-#2; 计算刀具球心 Z 轴坐标
G01 X#24 Z-#26 F#7; 直线切削到加工轮廓起始点
Y-2; 轮廓加工
#15=#16-#6; 角度变量#15 递减一个#6
END2; 循环 2 结束
N20 G01 X#30 Z#2; 序号 20 程序段: 直线切削到倒圆顶面
```

Y[#3+2]; 轮廓切削（前面数十次往复切削结束点有两种可能，所以这里安排往复切削，就是防止最后未切削）

Y-2; 轮廓切削

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

O567;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G52 Y-5; 建立局部坐标系，以 (0, -5, 0) 为原点

G65 P6567 A35 B8 C10 J5 K5 D300; 调用宏程序 O6567, 给相关变量赋初始值

G51.1 X0; 建立以 Y 轴为镜像轴的镜像坐标系

G65 P6567 A35 B8 C10 J5 K5 D300; 调用宏程序 O6567, 给相关变量赋初始值

G50.1 X0; 取消以 Y 轴为镜像轴的镜像坐标系

G52 Y0; 取消局部坐标系, 恢复 G54 坐标系

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

6.5.7 实例七椭圆柱倒圆

1. 外椭圆柱上平面倒圆

(1) 立铣刀

1) 分析: 如图 6-94 所示, X、Y 轴方向半轴长分别为 #1、#3 的外椭圆柱上平面为 XY 轴所在的平面, 中心为坐标系原点, 轴线为 Z 轴, 上平面四周需用直径为 #2 的立铣刀进行倒圆, R 为 #6。

走刀方式: 下刀点即开始点选择在工件的正前方, 由下而上逐层环铣, 以切入点与圆角圆心连线跟水平面之间的夹角为角度自变量 #8, #6 为角度递增量, 顺铣方式 (顺时针方向走刀铣削) 单向走刀。

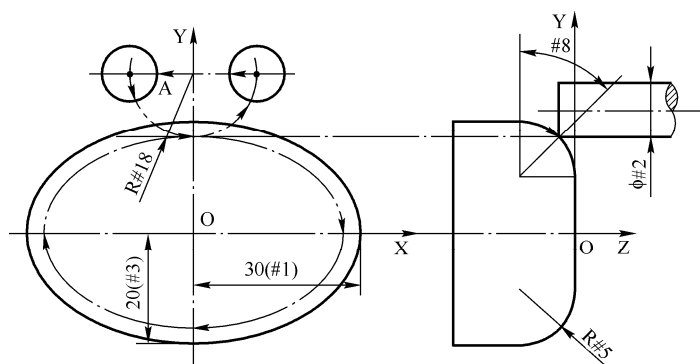


图 6-94 外椭圆柱上平面倒圆 (1)

首先快速定位在 $(0, [\#3+\#18], -\#5)$ 处, 赋自变量初始值 $\#8=0$, 计算此时刀具半径补偿值 $\#11$, 通过 `G10 L12 P1 R#11` 指令将半径补偿值 $\#11$ 赋给半径补偿值 $D1$ 寄存器, 然后再运用半径补偿指令 `G41 D1` 和圆弧切入、顺时针走刀切削水平椭圆轮廓、切出和取消半径补偿, 从而完成第一层的切削。然后角度自变量 $\#8$ 递增一个角度递增量 $\#6$, 再计算刀具半径补偿值 $\#11$ 并赋给半径补偿值 $D1$ 寄存器, 进而如第一层方法进行第二层切削, 一层一层直至完成全部倒圆切削加工。

这里水平椭圆轮廓的加工, 始终以 $\#1$ 、 $\#3$ 为 X 、 Y 方向的半轴长, 只是通过 `G10 L12 P1 R#11` 指令对角度变量 $\#8$ 的变化, 将相应的半径补偿值 $\#11$ 及时赋给半径补偿值 $D1$ 寄存器, 利用不同的半径补偿值来调整每层水平椭圆轮廓的大小, 这样简化了椭圆轮廓加工程序的编程难度。但要注意半径补偿值 $\#11$ 计算要准确, 能正确反映随着角度变量 $\#8$ 的变化刀具实际加工位置的变化。

2) 变量及说明:

$\#1=[A]$; 椭圆 X 轴向半轴长

$\#2=[B]$; 刀具直径

$\#3=[C]$; 椭圆 Y 轴向半轴长

$\#5=[J]$; 圆角半径 r

$\#6=[K]$; 角度递增值

$\#7=[D]$; 进给量

$\#18=[R]$; $1/4$ 圆弧切入切出半径

3) 宏程序如下:

%

O6571; 子程序号

G00 X0 Y $[\#3+\#18]$; 快速定位到加工准备位置

Z-#5; 快速下降到加工起始深度
#8=0; 给角度变量#8 赋初始值 0
WHILE[#8LE90] DO2; 循环 2 设定条件: 角度变量#8 不大于 90°
#11=#2/2-#5*[1-COS[#8]]; 计算对应于角度变量#8 变化的半径补偿值#11
#26=#5*[1-SIN[#8]]; 计算对应于角度变量#8 变化的背吃刀量#26
G10 L12 P1 R#11; 将半径补偿值#11 赋给半径补偿值寄存器 D1
G00 Z-#26; 快速定位到当前背吃刀量#26
G41 D1 X-#18 Y[#3+#18]; 建立半径补偿
G03 X0 Y#3 R#18 F#7; 圆弧切入
#16=90; 角度变量#16 赋初始值 90°
WHILE[#16GE-270] DO1; 循环 1 设定条件: 角度变量#16 不小于-270°
#24=#1*COS[#16]; 计算对应于角度变量#16 变化的 X 轴坐标值
#25=#3*SIN[#16]; 计算对应于角度变量#16 变化的 Y 轴坐标值
G01 X#24 Y#25; 直线切削到刚刚计算的切削点 (#24, #25)
#16=#16-#6; 角度变量#16 递减一个角度递增量#6
END1; 循环 1 结束
G03 X#18 Y[#3+#18] R#18; 圆弧切出
G40 G0 X0; 取消半径补偿
#8=#8+#6; 角度变量#8 递增一个角度递增量#6
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50 处
M99; 子程序结束
%
4) 主程序如下:
%
O571;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6571 A30 B16 C20 J10 K5 D1000 R10; 调用宏程序 O6571,
给相关变量赋初始值
G00 G49 Z100;
X0 Y150;
M05;
M30;

%

(2) 球头铣刀

1) 分析: 如图 6-95 所示, 与图 6-94 基本相同, 只不过刀具是半径为#2 的球头铣刀, 其余完全一样。

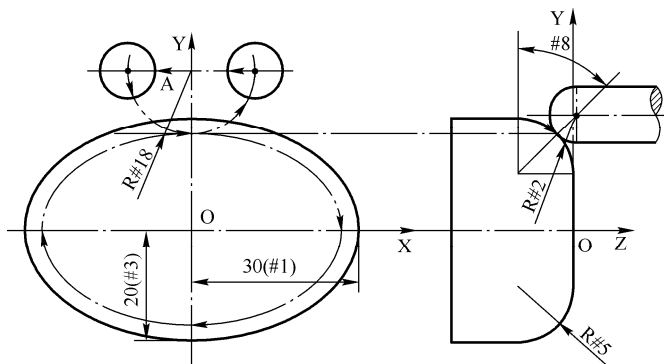


图 6-95 外椭圆柱上平面倒圆 (2)

走刀方式: 与图 6-94 完全一样。

首先快速定位在 $(0, [\#3+\#18], -\#5)$ 处, 赋自变量初始值 $\#8=0$, 这里没用半径补偿, 而是计算随着角度变量 $\#8$ 的变化, 倒圆后椭圆轮廓缩小量以及相应 X、Y 方向半轴长, 仍然进行圆弧切入、顺时针走刀切削水平椭圆轮廓、切出, 完成每一层的切削。然后角度自变量 $\#8$ 递增一个角度递增量 $\#6$, 再重新计算切削加工, 一层一层直至完成全部倒圆切削加工。

这里水平椭圆轮廓的加工, 以能反映角度变量 $\#8$ 变化的变量 $\#14$ 、 $\#15$ 作为 X、Y 方向的半轴长。

2) 变量及说明:

$\#1=[A]$; 椭圆 X 轴向半轴长

$\#2=[B]$; 球头铣刀半径

$\#3=[C]$; 椭圆 Y 轴向半轴长

$\#4=[I]$; 椭圆柱高度

$\#5=[J]$; 圆角半径 r

$\#6=[K]$; 角度递增值

$\#7=[D]$; 进给量

$\#18=[R]$; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6572; 子程序号

G00 X0 Y[#3+#18]; 快速定位到加工准备位置
Z-#5; 快速下降到加工起始深度
#8=0; 给角度变量#8 赋初始值 0
WHILE[#8LE90] DO2; 循环 2 设定条件: 角度变量#8 不大于 90°
#14=#1+[#2+#5]*COS[#8]-#5; 计算对应于角度变量#8 变化的 X 轴半轴长
#15=#3+[#2+#5]*COS[#8]-#5; 计算对应于角度变量#8 变化的 Y 轴半轴长
#26=[#2+#5]*[1-SIN[#8]]; 计算对应于角度变量#8 变化的背吃刀量#26
G00 Z-#26; 快速定位到当前背吃刀量#26
X-#18 Y[#15+#18]; 快速定位到圆弧切入起始点
G03 X0 Y#15 R#18 F#7; 圆弧切入
#16=90; 角度变量#16 赋初始值 90°
WHILE[#16GE-270] DO1; 循环 1 设定条件: 角度变量#16 不小于-270°
#24=#14*COS[#16]; 计算对应于角度变量#16 变化的 X 轴坐标值
#25=#15*SIN[#16]; 计算对应于角度变量#16 变化的 Y 轴坐标值
G01 X#24 Y#25; 直线切削到刚刚计算的切削点 (#24, #25)
#16=#16-#6; 角度变量#16 递减一个角度递增量#6
END1; 循环 1 结束
G03 X#18 Y[#15+#18] R#18; 圆弧切出
G0 X0; 返回加工起始点
#8=#8+#6; 角度变量#8 递增一个角度递增量#6
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50 处
M99; 子程序结束
%
4) 主程序如下:
%
O672;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6572 A30 B8 C20 J10 K5 D1000 R10; 调用宏程序 O6572,
给相关变量赋初始值
G00 G49 Z100;
X0 Y150;
M05;

M30;

%

2. 外椭圆柱底面倒圆

(1) 立铣刀

1) 分析: 如图 6-96 所示, X、Y 轴方向半轴长分别为#1、#3 的外椭圆柱上平面为 XY 轴所在的平面, 中心为坐标系原点, 轴线为 Z 轴, 上平面四周需用直径为#2 的立铣刀进行倒圆, R 为#6。

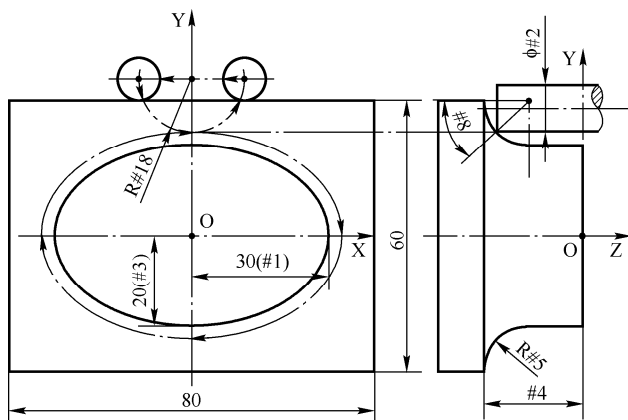


图 6-96 外椭圆柱底面倒圆 (1)

走刀方式: 下刀点即开始点选择在工件的正前方, 由下而上逐层环铣, 以切入点与圆角圆心连线跟垂直面之间的夹角为角度自变量#8, #6 为角度递增量, 顺铣方式(顺时针方向走刀铣削)单向走刀。

首先快速定位在 $(0, [\#3 + \#5 + \#18], -\#4)$ 处, 赋自变量初始值#8=0, 计算此时刀具半径补偿值#11, 通过 G10 L12 P1 R#11 指令将半径补偿值 #11 赋给半径补偿值 D1 寄存器, 然后再运用半径补偿指令 G41 D1 和圆弧切入、顺时针走刀切削水平椭圆轮廓、切出和取消半径补偿, 从而完成第一层的切削。然后角度自变量#8 递增一个角度递增量#6, 再计算刀具半径补偿值#11 并赋给半径补偿值 D1 寄存器, 进而如第一层方法进行第二层切削, 一层一层直至完成全部倒圆切削加工。

这里水平椭圆轮廓的加工, 始终以#1、#3 为 X、Y 方向的半轴长, 只是通过 G10 L12 P1 R#11 指令对角度变量#8 的变化, 将相应的半径补偿值#11 及时赋给半径补偿值 D1 寄存器, 利用不同的半径补偿值来调整每层水平椭圆轮廓的大小, 这样简化了椭圆轮廓加工程序的编程难度。但要注意半径补偿值#11 计算要准确, 能正确反映随着角度变量#8 的变化刀具实际加工位置的变化。

2) 变量及说明:

#1=[A]; 椭圆 X 轴向半轴长

#2=[B]; 刀具直径

#3=[C]; 椭圆 Y 轴向半轴长

#5=[J]; 圆角半径 r

#6=[K]; 角度递增值

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6573; 子程序号

G00 X0 Y[#3+#5+#18]; 快速定位到加工准备位置

Z-#5; 快速下降到加工起始深度

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO2; 循环 2 设定条件: 角度变量#8 不大于 90°

#11=#2/2+#5*[1-SIN[#8]]; 计算对应于角度变量#8 变化的半径补偿值#11

#26=#4+#5*[COS[#8]-1]; 计算对应于角度变量#8 变化的背吃刀量#26

G10 L12 P1 R#11; 将半径补偿值#11 赋给半径补偿值寄存器 D1

G00 Z-#26; 快速定位到当前背吃刀量#26

G41 D1 X-#18 Y[#3+#18]; 建立半径补偿

G03 X0 Y#3 R#18 F#7; 圆弧切入

#16=90; 角度变量#16 赋初始值 90°

WHILE[#16GE-270] DO1; 循环 1 设定条件: 角度变量#16 不小于-270°

#24=#1*COS[#16]; 计算对应于角度变量#16 变化的 X 轴坐标值

#25=#3*SIN[#16]; 计算对应于角度变量#16 变化的 Y 轴坐标值

G01 X#24 Y#25; 直线切削到刚刚计算的切削点 (#24, #25)

#16=#16-#6; 角度变量#16 递减一个角度递增量#6

END1; 循环 1 结束

G03 X#18 Y[#3+#18] R#18; 圆弧切出

G40 G0 X0 Y[#3+#5+#18]; 取消半径补偿

#8=#8+#6; 角度变量#8 递增一个角度递增量#6

END2; 循环 2 结束

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

```

%
O673;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6573 A30 B16 C20 I25 J10 K5 D1000 R10; 调用宏程序 O6573,
                                             给相关变量赋初始值

G00 G49 Z100;
X0 Y150;
M05;
M30;
%
```

(2) 球头铣刀

1) 分析: 如图 6-97 所示, 与图 6-96 基本相同, 只不过刀具是半径为#2 的球头铣刀, 其余完全一样。

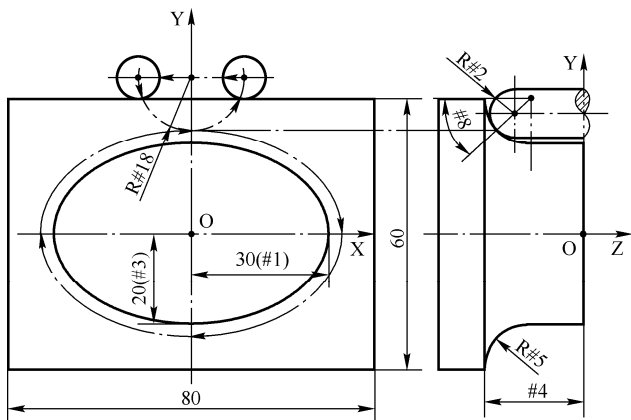


图 6-97 外椭圆柱底面倒圆 (2)

走刀方式: 与图 6-96 完全一样。

首先快速定位在 $(0, [\#3+\#5+\#18], [-\#4+\#2])$ 处, 赋自变量初始值#8=0, 这里没用半径补偿, 而是计算随着角度变量#8 的变化, 倒圆后椭圆轮廓缩小量以及相应 X、Y 方向半轴长, 仍然进行圆弧切入、顺时针走刀切削水平椭圆轮廓、切出, 完成每一层的切削。然后角度自变量#8 递增一个角度递增量#6, 再重新计算切削加工, 一层一层直至完成全部倒圆切削加工。

这里水平椭圆轮廓的加工, 以能反映角度变量#8 变化的变量#14、#15 作为 X、

Y 方向的半轴长。

2) 变量及说明:

#1=[A]; 椭圆 X 轴向半轴长

#2=[B]; 球头铣刀半径

#3=[C]; 椭圆 Y 轴向半轴长

#4=[I]; 椭圆柱高度

#5=[J]; 圆角半径 r

#6=[K]; 角度递增值

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6574; 子程序号

G00 X0 Y[#3+#5+#18]; 快速定位到加工准备位置

Z-[#4-#2-2]; 快速下降到离加工起始深度 2mm 处

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO2; 循环 2 设定条件: 角度变量#8 不大于 90°

#14=#1-[#6-#2]*SIN[#8]+#5; 计算对应于角度变量#8 变化的 X 轴半轴长

#15=#3-[#6-#2]*SIN[#8]+#5; 计算对应于角度变量#8 变化的 Y 轴半轴长

#26=#4-#5+[#6-#2]*COS[#8]; 计算对应于角度变量#8 变化的背吃刀量#26

G01 Z-#26 F#7; 直线切削到当前背吃刀量#26

X-#18 Y[#15+#18]; 直线切削到圆弧切入起始位置

G03 X0 Y#15 R#18; 圆弧切入

#16=90; 角度变量#16 赋初始值 90°

WHILE[#16GE-270] DO1; 循环 1 设定条件: 角度变量#16 不小于-270°

#24=#1*COS[#16]; 计算对应于角度变量#16 变化的 X 轴坐标值

#25=#3*SIN[#16]; 计算对应于角度变量#16 变化的 Y 轴坐标值

G01 X#24 Y#25; 直线切削到刚刚计算的切削点 (#24, #25)

#16=#16-#6; 角度变量#16 递减一个角度递增量#6

END1; 循环 1 结束

G03 X#18 Y[#15+#18] R#18; 圆弧切出

G01 X0[#3+#5+#18]; 直线切削到加工起始位置

#8=#8+#6; 角度变量#8 递增一个角度递增量#6

END2; 循环 2 结束

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

O674;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6574 A30 B8 C20 I25 J10 K5 D1000 R10; 调用宏程序 O6574,
给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

3. 内椭圆槽孔口倒圆

(1) 立铣刀

1) 分析: 如图 6-98 所示, X、Y 轴方向半轴长分别为 #1、#3 的内椭圆槽上平面为 X、Y 轴所在的平面, 中心为坐标系原点, 轴线为 Z 轴, 槽底面四周需用直径为 #2 的立铣刀进行倒圆, R 为 #6。

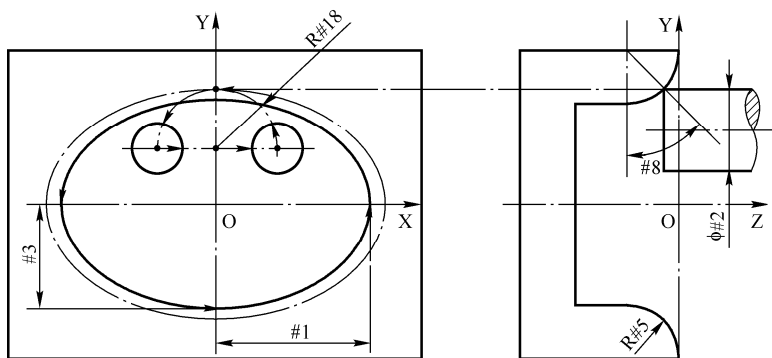


图 6-98 内椭圆槽孔口倒圆 (1)

走刀方式: 下刀点即开始点选择在槽的正前方偏后, 由底面而上逐层环铣, 以切入点与圆角圆心连线跟垂直面之间的夹角为角度自变量 #8, #6 为角度递增量, 顺铣方式 (逆时针方向走刀铣削) 单向走刀。

首先快速定位在 (0, [#3-#18], -#5) 处, 赋自变量初始值 #8=0, 计算此时刀

具半径补偿值#11,通过 G10 L12 P1 R#11 指令将半径补偿值 #11 赋给半径补偿值 D1 寄存器,然后再运用半径补偿指令 G41 D1 和圆弧切入、逆时针走刀切削水平椭圆轮廓、切出和取消半径补偿,从而完成第一层的切削。然后角度自变量#8 递增一个角度递增量#6,再计算刀具半径补偿值#11 并赋给半径补偿值 D1 寄存器,进而如第一层方法进行第二层切削,一层一层直至完成全部倒圆切削加工。

这里水平椭圆轮廓的加工,始终以#1、#3 为 X、Y 方向的半轴长,只是通过 G10 L12 P1 R#11 指令对角度变量#8 的变化,将相应的半径补偿值#11 及时赋给半径补偿值 D1 寄存器,利用不同的半径补偿值来调整每层水平椭圆轮廓的大小,这样简化了椭圆轮廓加工程序的编程难度。但要注意半径补偿值#11 计算要准确,能正确反映随着角度变量#8 的变化刀具实际加工位置的变化。

2) 变量及说明:

#1=[A]; 椭圆 X 轴向半轴长

#2=[B]; 刀具直径

#3=[C]; 椭圆 Y 轴向半轴长

#5=[J]; 圆角半径 r

#6=[K]; 角度递增值

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6575; 子程序号

G00 X0 Y[#3-#6-#18]; 快速定位到加工准备位置

Z-#5; 快速下降到加工起始深度

#8=0; 给角度变量#8 赋初始值 0

WHILE[#8LE90] DO2; 循环 2 设定条件: 角度变量#8 不大于 90°

#11=#2/2-#5*[1-COS[#8]]; 计算对应于角度变量#8 变化的半径补偿值#11

#26=#6-#5*SIN[#8]; 计算对应于角度变量#8 变化的背吃刀量#26

G10 L12 P1 R#11; 将半径补偿值#11 赋给半径补偿值寄存器 D1

G41 D1 X#18 Y[#3-#18]; 建立半径补偿

G03 X0 Y#3 Z#-26 R#18 F#7; 螺旋切入

#16=90; 角度变量#16 赋初始值 90°

WHILE[#16LE450] DO1; 循环 1 设定条件: 角度变量#16 不大于 450°

#24=#1*COS[#16]; 计算对应于角度变量#16 变化的 X 轴坐标值

#25=#3*SIN[#16]; 计算对应于角度变量#16 变化的 Y 轴坐标值

G01 X#24 Y#25; 直线切削到刚刚计算的切削点 (#24, #25)

#16=#16+#6; 角度变量#16 递增一个角度递增量#6

END1; 循环 1 结束

G03 X-#18 Y[#3-#18] Z[2-#26] R#18; 螺旋切出

G40 G0 X0 Y[#3-#6-#18]; 取消半径补偿

#8=#8+#6; 角度变量#8 递增一个角度递增量#6

END2; 循环 2 结束

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

O575;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6575 A40 B16 C30 J10 K5 D1000 R10; 调用宏程序 O6575,
给相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

(2) 球头铣刀

1) 分析: 如图 6-99 所示, 与图 6-98 基本相同, 只不过刀具是半径为#2 的球头铣刀, 其余完全一样。

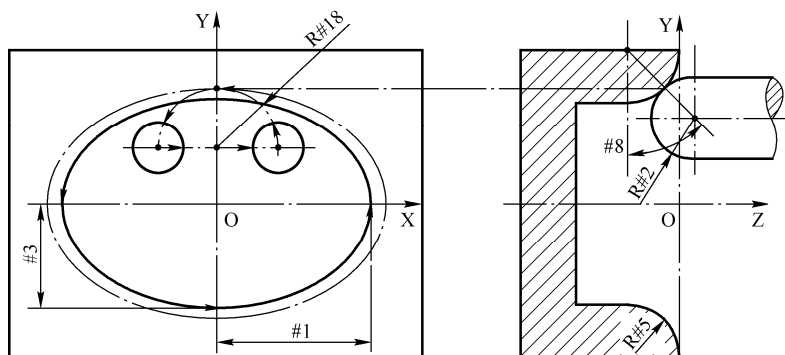


图 6-99 内椭圆槽孔口倒圆 (2)

走刀方式：与图 6-98 完全一样。

首先快速定位在 (0, [#3-#18], -#5) 处，赋自变量初始值#8=0，这里没用半径补偿，而是计算随着角度变量#8 的变化，倒圆后椭圆轮廓增大量以及相应 X、Y 方向半轴长，仍然进行圆弧切入、逆时针走刀切削水平椭圆轮廓、切出，完成每一层的切削。然后角度自变量#8 递增一个角度递增量#6，再重新计算切削加工，一层一层直至完成全部倒圆切削加工。

这里水平椭圆轮廓的加工，以能反映角度变量#8 变化的变量#14、#15 作为 X、Y 方向的半轴长。

2) 变量及说明：

#1=[A]；椭圆 X 轴向半轴长

#2=[B]；球头铣刀半径

#3=[C]；椭圆 Y 轴向半轴长

#5=[J]；圆角半径 r

#6=[K]；角度递增值

#7=[D]；进给量

#18=[R]；1/4 圆弧切入切出半径

3) 宏程序如下：

%

O6576；子程序号

G00 X0 Y[#3-#6-#18]；快速定位到加工准备位置

Z-#5；快速下降到加工起始深度

#8=0；给角度变量#8 赋初始值 0

WHILE[#8LE90] DO2；循环 2 设定条件：角度变量#8 不大于 90°

#14=#1-[#5+#2]*COS[#8]+#5；计算对应于角度变量#8 变化的 X 轴半轴长

#15=#3-[#5+#2]*COS[#8]+#5；计算对应于角度变量#8 变化的 Y 轴半轴长

#26=#6-[#5+#2]*SIN[#8]；计算对应于角度变量#8 变化的背吃刀量#26

G00 X#18 Y[#16-#18]；快速定位到螺旋切入起始点

G03 X0 Y#15 Z-#26 R#18 F#7；螺旋切入

#16=90；角度变量#16 赋初始值 90°

WHILE[#16LE450] DO1；循环 1 设定条件：角度变量#16 不大于 450°

#24=#14*COS[#16]；计算对应于角度变量#16 变化的 X 轴坐标值

#25=#15*SIN[#16]；计算对应于角度变量#16 变化的 Y 轴坐标值

G01 X#24 Y#25；直线切削到刚刚计算的切削点 (#24, #25)

#16=#16-#6；角度变量#16 递减一个角度递增量#6

END1；循环 1 结束

G03 X-#18 Y[#16-#18] Z[2-#26] R#18; 螺旋切出

G00 X0 Y[#3-#6-#18]; 快速返回加工准备位置

#8=#8+#6; 角度变量#8 递增一个角度递增量#6

END2; 循环 2 结束

G00 Z50; 快速上升到 Z=50 处

M99; 子程序结束

%

4) 主程序如下:

%

O576;

G54 G90 G00 G43 H1 Z50;

X0 Y0;

M03 S1000;

G65 P6576 A40 B8 C30 J10 K5 D1000 R10; 调用宏程序 O6576, 给
相关变量赋初始值

G00 G49 Z100;

X0 Y150;

M05;

M30;

%

4. 内椭圆槽底面倒圆

(1) 立铣刀

1) 分析: 如图 6-100 所示, X、Y 轴方向半轴长分别为#1、#3 的内椭圆槽上平面为 XY 轴所在的平面, 中心为坐标系原点, 轴线为 Z 轴, 槽底面四周需用直径为#2 的立铣刀进行倒圆, R 为#6。

走刀方式: 下刀点即开始点选择在槽的正前方偏后, 由底面而上逐层环铣, 以切入点与圆角圆心连线跟垂直面之间的夹角为角度自变量#8, #6 为角度递增量, 顺铣方式(逆时针方向走刀铣削)单向走刀。

首先快速定位在(0, [#3-#18], -#4)处, 赋自变量初始值#8=0, 计算此时刀具半径补偿值#11, 通过 G10 L12 P1 R#11 指令将半径补偿值#11 赋给半径补偿值 D1 寄存器, 然后再运用半径补偿指令 G41 D1 和圆弧切入、逆时针走刀切削水平椭圆轮廓、切出和取消半径补偿, 从而完成第一层的切削。然后角度自变量#8 递增一个角度递增量#6, 再计算刀具半径补偿值#11 并赋给半径补偿值 D1 寄存器, 进而如第一层方法进行第二层切削, 一层一层直至完成全部倒圆切削加工。

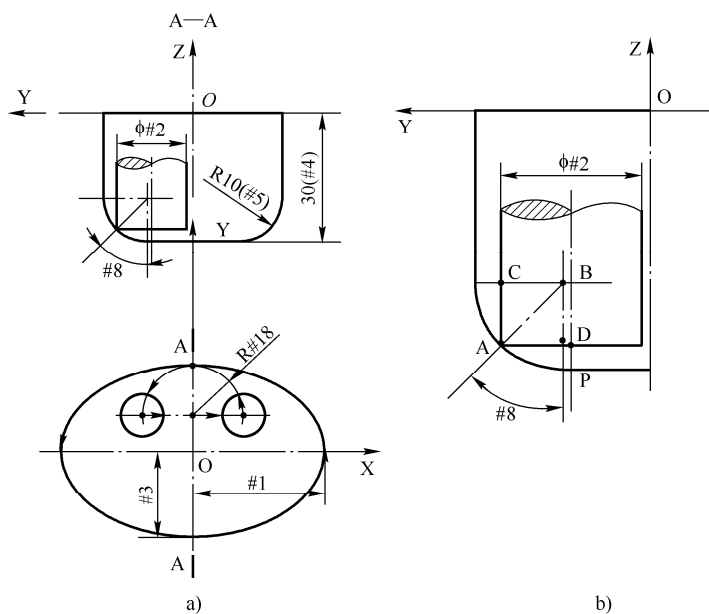


图 6-100 内椭圆槽底面倒圆 (1)

a) 零件示意图 b) 局部放大图

这里水平椭圆轮廓的加工, 始终以#1、#3为X、Y方向的半轴长, 只是通过G10 L12 P1 R#11指令对角度变量#8的变化, 将相应的半径补偿值#11及时赋给半径补偿值D1寄存器, 利用不同的半径补偿值来调整每层水平椭圆轮廓的大小, 这样简化了椭圆轮廓加工程序的编程难度。但要注意半径补偿值#11计算要准确, 能正确反映随着角度变量#8的变化刀具实际加工位置的变化。

2) 变量及说明:

#1=[A]; 椭圆X轴向半轴长

#2=[B]; 刀具直径

#3=[C]; 椭圆Y轴向半轴长

#4=[I]; 槽深度

#5=[J]; 圆角半径r

#6=[K]; 角度递增值

#7=[D]; 进给量

#18=[R]; 1/4圆弧切入切出半径

3) 宏程序如下:

%

O6577; 子程序号

G00 X0 Y[#3-#6-#18]; 快速定位到加工准备位置
Z[2-#4]; 快速下降到离加工起始深度 2mm 处
#8=0; 给角度变量#8 赋初始值 0
WHILE[#8LE90] DO2; 循环 2 设定条件: 角度变量#8 不大于 90°
#11=#2/2+#5*[1-SIN[#8]]; 计算对应于角度变量#8 变化的半径补偿值#11
#26=#4-#5+#5*COS[#8]; 计算对应于角度变量#8 变化的背吃刀量#26
G10 L12 P1 R#11; 将半径补偿值#11 赋给半径补偿值寄存器 D1
G41 D1 X#18 Y[#16-#18]; 建立半径补偿
G03 X0 Y#3 Z#-26 R#18 F#7; 螺旋切入
#16=90; 角度变量#16 赋初始值 90°
WHILE[#16LE450] DO1; 循环 1 设定条件: 角度变量#16 不大于 450°
#24=#1*COS[#16]; 计算对应于角度变量#16 变化的 X 轴坐标值
#25=#3*SIN[#16]; 计算对应于角度变量#16 变化的 Y 轴坐标值
G01 X#24 Y#25; 直线切削到刚刚计算的切削点 (#24, #25)
#16=#16-#6; 角度变量#16 递减一个角度递增量#6
END1; 循环 1 结束
G03 X-#18 Y[#3-#18] Z[2-#26] R#18; 螺旋切出
G40 G0 X0 Y[#3-#6-#18]; 取消半径补偿
#8=#8+#6; 角度变量#8 累加一个角度递增量#6
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50 处
M99; 子程序结束
%
4) 主程序如下:
%
O577;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6577 A40 B16 C30 I25 J10 K5 D1000 R10; 调用宏程序 O6577,
给相关变量赋初始值
G00 G49 Z100;
X0 Y150;
M05;
M30;

%

(2) 球头铣刀

1) 分析: 如图 6-101 所示, 与图 6-100 基本相同, 只不过刀具是半径为#2 的球头铣刀, 其余完全一样。

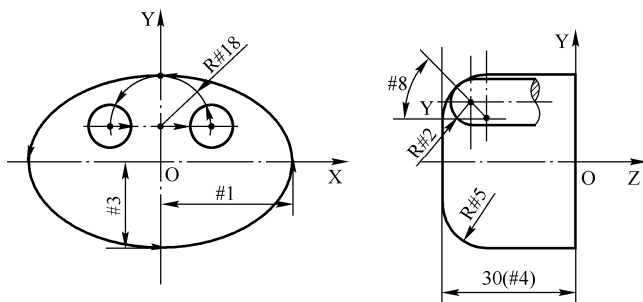


图 6-101 内椭圆槽底面倒圆 (2)

走刀方式: 与图 6-100 完全一样。

首先快速定位在 $(0, [\#3-\#6-\#18], -\#4)$ 处, 赋自变量初始值#8=0, 这里没用半径补偿, 而是计算随着角度变量#8 的变化, 倒圆后椭圆轮廓增量以及相应 X、Y 方向半轴长, 仍然进行圆弧切入、逆时针走刀切削水平椭圆轮廓、切出, 完成每一层的切削。然后角度自变量#8 递增一个角度递增量#6, 再重新计算切削加工, 一层一层直至完成全部倒圆切削加工。

这里水平椭圆轮廓的加工, 以能反映角度变量#8 变化的变量#14、#15 作为 X、Y 方向的半轴长。

2) 变量及说明:

#1=[A]; 椭圆 X 轴向半轴长

#2=[B]; 球头铣刀半径

#3=[C]; 椭圆 Y 轴向半轴长

#4=[I]; 槽深度

#5=[J]; 圆角半径 r

#6=[K]; 角度递增值

#7=[D]; 进给量

#18=[R]; 1/4 圆弧切入切出半径

3) 宏程序如下:

%

O6578; 子程序号

G00 X0 Y[#3-#6-#18]; 快速定位到加工准备位置

Z[2-#4]; 快速下降到离加工起始深度 2mm 处
#8=0; 给角度变量#8 赋初始值 0
WHILE[#8LE90] DO2; 循环 2 设定条件: 角度变量#8 不大于 90°
#14=#1+[#6-#2]*SIN[#8]-#5; 计算对应于角度变量#8 变化的 X 轴半轴长
#15=#3+[#6-#2]*SIN[#8]-#5; 计算对应于角度变量#8 变化的 Y 轴半轴长
#26=#4-#5+[#6-#2]*COS[#8]; 计算对应于角度变量#8 变化的背吃刀量#26
X#18 Y[#16-#18]; 快速定位到螺旋切入起始点
G03 X0 Y#15 Z-#26 R#18 F#7; 螺旋切入
#16=90; 角度变量#16 赋初始值 90°
WHILE[#16LE450] DO1; 循环 1 设定条件: 角度变量#16 不大于 450°
#24=#1*COS[#16]; 计算对应于角度变量#16 变化的 X 轴坐标值
#25=#3*SIN[#16]; 计算对应于角度变量#16 变化的 Y 轴坐标值
G01 X#24 Y#25; 直线切削到刚刚计算的切削点 (#24, #25)
#16=#16-#6; 角度变量#16 递减一个角度递增量#6
END1; 循环 1 结束
G03 X-#18 Y[#16-#18] Z[2-#26] R#18; 螺旋切出
G00 X0 Y[#3-#6-#18]; 快速返回加工准备位置
#8=#8+#6; 角度变量#8 递增一个角度递增量#6
END2; 循环 2 结束
G00 Z50; 快速上升到 Z=50 处
M99; 子程序结束
%
4) 主程序如下:
%
O578;
G54 G90 G00 G43 H1 Z50;
X0 Y0;
M03 S1000;
G65 P6578 A40 B8 C30 I25 J10 K5 D1000 R10; 调用宏程序 O6578,
给相关变量赋初始值
G00 G49 Z100;
X0 Y150;
M05;
M30;
%

附录

附录 A FANUC 0i 系统常用准备功能代码

G 代码	模态	功 能	G 代码	模态	功 能
*G00	01	定位（快速移动）	G57	14	选择工件坐标系 4
G01	01	直线切削	G58	14	选择工件坐标系 5
G02	01	圆弧插补（CW，顺时针）	G59	14	选择工件坐标系 6
G03	01	圆弧插补（CCW，逆时针）	G70	00	精加工循环
G04	00	暂停	G71	00	内外径粗切循环
G09	00	停于精确的位置	G72	00	台阶粗切循环
G20	06	英制输入	G73	00	成形重复循环
G21	06	米制输入	G74	00	Z 向进给钻削
G22	04	内部行程限位 有效	G75	00	X 向切槽
G23	04	内部行程限位 无效	G76	00	切螺纹循环
G27	00	检查参考点返回	*G80	10	固定循环取消
G28	00	参考点返回	G83	10	钻孔循环
G29	00	从参考点返回	G84	10	攻丝循环
G30	00	回到第二参考点	G85	10	正面镗循环
G32	01	切螺纹	G87	10	侧钻循环
*G40	07	取消刀尖半径偏置	G88	10	侧攻丝循环
G41	07	刀尖半径偏置（左侧）	G89	10	侧镗循环
G42	07	刀尖半径偏置（右侧）	G90	01	（内外直径）切削循环
G50	00	主轴最高转速设置（坐标系设定）	G92	01	切螺纹循环
G52	00	设置局部坐标系	G94	01	（台阶）切削循环
G53	00	选择机床坐标系	G96	12	恒线速度控制
*G54	14	选择工件坐标系 1	*G97	12	恒线速度控制取消
G55	14	选择工件坐标系 2	G98	05	指定每分钟移动量
G56	14	选择工件坐标系 3	*G99	05	指定每转移动量

注：*表示系统开机默认状态。

附录 B FANUC 0i –TC 系统常用辅助功能代码

M 代码	功 能	M 代码	功 能
M00	程序停止	M41	主轴齿轮在低速位置
M01	选择性程序停止	M42	主轴齿轮在高速位置
M02	程序结束	M68	液压卡盘夹紧
M30	程序结束复位	M69	液压卡盘松开
M03	主轴正转	M78	尾架前进
M04	主轴反转	M79	尾架后退
M05	主轴停止	M94	镜像取消
M08	切削液启动	M95	X 坐标镜像
M09	切削液停	M98	子程序调用
M40	主轴齿轮在中间位置	M99	子程序结束

附录 C FANUC 0i –MC 系统常用准备功能代码

G 代码	模态	功 能	G 代码	模态	功 能
G00	01	点定位	G30	00	返回第 2、3、4 参考点
G01	01	直线插补	G31	00	跳转功能
G02	01	顺圆弧插补/螺旋线插补 CW	G33	01	螺纹切削
G03	01	逆圆弧插补/螺旋线插补 CCW	G37	00	自动刀具长度测量
G04	00	暂停、准确停止	G39	00	拐角偏置圆弧插补
G05.1	00	预读控制（超前读多程序段）	G40	07	刀具半径补偿取消
G07	00	圆柱插补	G41	07	刀具半径补偿，左侧
G08	00	预读控制	G42	07	刀具半径补偿，右侧
G09	00	准确停止	G40.1	18	法线方向控制取消方式
G10	00	可编程数据输入	G41.1	18	法线方向控制左侧接通
G11	00	可编程数据输入方式取消	G42.1	18	法线方向控制右侧接通
G15	17	极坐标指令消除	G43	08	正向刀具长度补偿
G16	17	极坐标指令	G44	08	负向刀具长度补偿
G17	02	选择 XY 平面	G45	00	刀具位置偏置加
G18	02	选择 XZ 平面	G46	00	刀具位置偏置减
G19	02	选择 YZ 平面	G47	00	刀具位置偏置加 2 倍
G20	06	英寸输入	G48	00	刀具位置偏置减一半
G21	06	毫米输入	G49	08	刀具长度补偿取消
G22	04	存储行程检测功能接通	G50	11	比例缩放取消
G23	04	存储行程检测功能断开	G51	11	比例缩放有效
G27	00	返回参考点检测	G50.1	22	可编程镜像取消
G28	00	返回参考点	G51.1	22	可编程镜像有效
G29	00	从参考点返回	G52	00	局部坐标系设定

(续)

G 代码	模态	功 能	G 代码	模态	功 能
G53	00	选择机床坐标系	G80	09	固定循环取消
G54	14	选择工件坐标系 1	G81	09	点钻循环
G54.1	14	选择附加工件坐标系	G82	09	镗孔循环
G55	14	选择工件坐标系 2	G83	09	深孔钻循环
G56	14	选择工件坐标系 3	G84	09	右旋攻丝循环
G57	14	选择工件坐标系 4	G85	09	镗孔循环
G58	14	选择工件坐标系 5	G86	09	镗孔循环
G59	14	选择工件坐标系 6	G87	09	背镗循环
G60	00/01	单方向定位	G88	09	镗孔循环
G61	15	准确停止方式	G89	09	镗孔循环
G62	15	自动拐角倍率	G90	03	绝对值编程
G63	15	攻丝方式	G91	03	增量值编程
G64	15	切削方式	G92	00	设定工件坐标系
G65	00	宏程序调用	G92.1	00	工件坐标系预置
G66	12	宏程序模态调用	G94	05	每分钟进给
G67	12	宏程序模态调用取消	G95	05	主轴每转进给
G68	16	坐标旋转有效	G96	13	恒线速控制
G69	16	坐标旋转取消	G97	13	恒线速控制取消
G73	09	深孔钻循环	G98	10	循环返回到初始点
G74	09	左旋攻丝循环	G99	10	循环返回到 R 点
G76	09	精镗循环			

附录 D 可变更加工坐标系指令 G10 简介

1. 格式: **G10 L10 P_R_**; 用于在程序中输入刀具的长度补偿值

其中:

P: 刀具的补偿号。

R: 刀具的补偿值。

当用 **G90** 绝对值指令方式时, **R** 后接的数值就是刀具的长度补偿值; 当用 **G91** 增量指令方式时, **R** 后接的数值和指定的刀具长度补偿值的和就是刀具的长度补偿值。

2. 格式: **G10 L12 P_R_**; 用于在程序中输入刀具的半径补偿值

其中:

P: 刀具的补偿号。

R: 刀具的补偿值。

当用 **G90** 绝对值指令方式时, **R** 后接的数值就是刀具的半径补偿值; 当用 **G91** 增量指令方式时, **R** 后接的数值和指定的刀具半径补偿值的和就是刀具的半径补偿值。

参 考 文 献

- [1] 陈海舟. 数控铣削加工宏程序及应用实例[M]. 北京: 机械工业出版社, 2006.
- [2] 《数控大赛试题·答案·点评》编委会. 数控大赛试题·答案·点评[M]. 北京: 机械工业出版社, 2006.
- [3] 李锋, 白一凡. 数控铣削变量编程实例教程[M]. 北京: 化学工业出版社, 2007.
- [4] 李锋. 数控宏程序实例教程[M]. 北京: 化学工业出版社, 2010.
- [5] 孙德茂. 数控机床铣削加工直接编程技术[M]. 北京: 机械工业出版社, 2004.

详细介绍宏程序基础知识

大量实例分析，迅速掌握技巧

赠送多媒体动画光盘，提高学习效果

ISBN 978-7-111-34860-3

ISBN 978-7-89451-982-5 (光盘)

策划编辑：周国萍

封面设计：路恩中

地址：北京市百万庄大街22号

电话服务

社服务中心：(010)88361066

销售一部：(010)68326294

销售二部：(010)88379649

读者购书热线：(010)88379203

邮政编码：100037

网络服务

门户网：<http://www.cmpbook.com>

教材网：<http://www.cmpedu.com>

封面无防伪标均为盗版

上架指导：工业技术 / 机械工程 / 数控技术

ISBN 978-7-111-34860-3



9 787111 348603 >

定价：48.00元 (含1CD)