

轻松掌握



FANUC宏程序

——编程技巧与实例精解

杜 军 编著

QINGSONG ZHANGWO FANUC HONGCHENGXU
BIANCHENG JIQIAO YU SHILI JINGJIE



化学工业出版社



FANUC宏程序

——编程技巧与实例精解

杜 军 编著



QINGSONG ZHANGWO FANUC HONGCHENGXU
BIANCHENG JIQIAO YU SHILI JINGJIE



化学工业出版社

· 北 京 ·

这是一本让你轻松实现从入门到精通 FANUC 数控宏程序编程的书籍。

本书是实用性非常强的数控技术用书,详细介绍了以 FANUC 0i 系统为蓝本的 B 类宏程序的基础知识、数控车削加工宏程序编程和数控铣削加工宏程序编程相关知识。全书内容采用“实例法”由浅入深,由易到难,循序渐进的模块化方式编写,共分 56 个模块,先介绍相关入门基础知识导入学习,然后精选 102 道典型例题详细讲解以期重难点突破,最后精心设计了 200 余道针对性思考练习题供强化练习巩固提高(附参考答案),完全符合科学的学习模式。

本书可供数控行业的工程技术人员、从事数控加工编程及操作人员使用,也可作为各类大中专院校或培训学校的数控相关专业师生使用,还可作为各类数控竞赛和国家职业技能鉴定数控高级工、数控技师、高级技师的参考书。

图书在版编目(CIP)数据

轻松掌握 FANUC 宏程序——编程技巧与实例精解/
杜军编著. —北京:化学工业出版社, 2011.1
ISBN 978-7-122-10102-0

I. 轻… II. ①杜… III. 数控机床-程序设计
IV. TG659

中国版本图书馆 CIP 数据核字 (2010) 第 241822 号

责任编辑:张兴辉
责任校对:周梦华

文字编辑:项 渤
装帧设计:王晓宇



出版发行:化学工业出版社(北京市东城区青年湖南街 13 号 邮政编码 100011)
印 装:大厂聚鑫印刷有限责任公司

787mm×1092mm 1/16 印张 15 字数 368 千字 2011 年 3 月北京第 1 版第 1 次印刷

购书咨询:010-64518888(传真:010-64519686) 售后服务:010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定 价:46.00 元

版权所有 违者必究

前言 FOREWORD

这是一本让你轻松实现从入门到精通 FANUC 数控宏程序编程的书籍。

如果你是数控编程学习人员，你是数控加工从业人员，那你懂宏程序吗？……不懂？……你会用宏程序吗？……不会？太难？……你对宏程序的应用了解全面吗？……了解一部分，不全面？……你 OUT（落伍）了！

什么是宏程序？

先看下面的例题：数控车削精加工图 1 所示“ $\phi 30 \times 40$ ”外圆柱面，设工件坐标系原点在右端面与轴线的交点上，加工路线 A→B，编制加工程序为“G01 X30 Z-40 F0.1”，其中“30”、“40”是常量。若将图 1 中直径尺寸“ $\phi 30$ ”和长度尺寸“40”分别用符号“D”和“L”替换（如图 2 所示），编制加工程序为“G01 XD Z-L F0.1”。

不用想太多，“D”和“L”都是符号而已，那么用其他符号代替也可以的吧。将图 2 中符号“D”和“L”分别用数控机床能认识的符号“#1”和“#2”替换（如图 3 所示，“#1”、“#2”就是变量），加工程序就变为“G01 X#1 Z-#2 F0.1”，这就是一个宏程序语句！

定义有了：含有“#i”（变量）符号的程序就叫宏程序！！很简单！

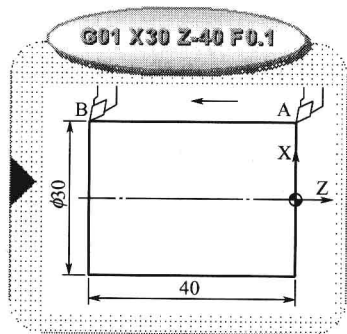


图 1

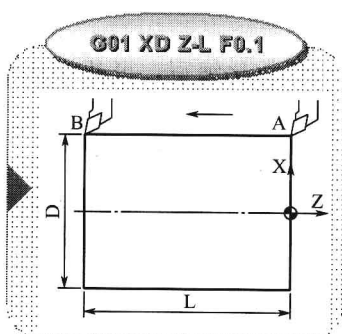


图 2

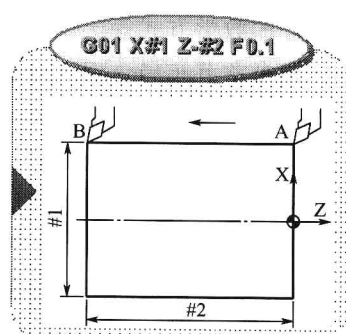


图 3

宏程序给你带来什么？

宏程序的典型应用：

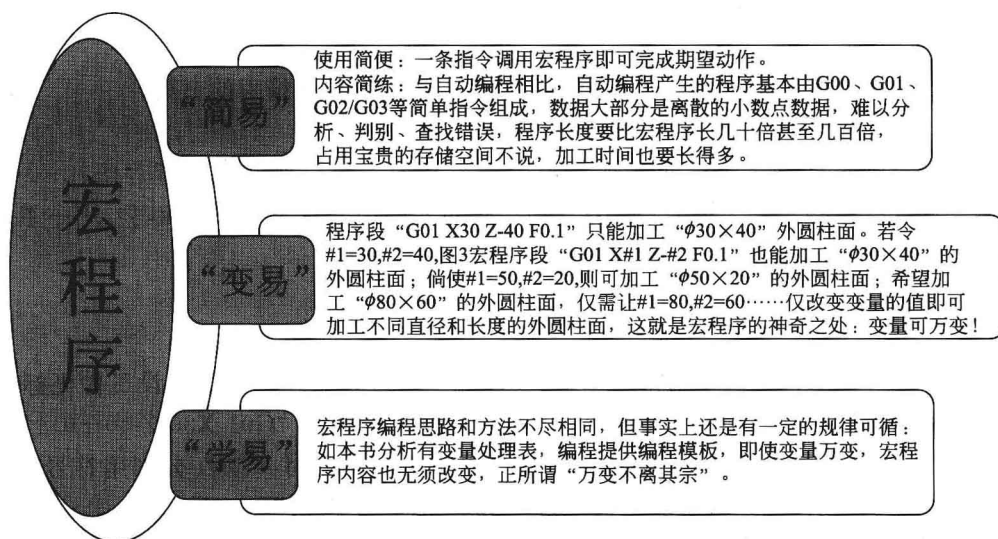
(1) 定制专属固定循环指令

重复出现的相同结构、形状相似、尺寸不同的系列产品，还有如大平面铣削等典型的循环动作均可轻松实现一条指令调用宏程序完成期望动作，将重复或复杂的问题简单化，还可设为 G、M、T 指令方便调用。

(2) 实现曲线插补

系统仅提供了直线插补和圆弧插补，运用宏程序可实现公式曲线的插补功能，椭圆、双曲线、抛物线、正弦曲线等非圆曲线尽在掌握，可大大拓展系统插补指令。

关于宏程序能给你带来的改变，也许从下图能有所了解。



本书为你提供什么？

(1) 教练式教学，学习轻松简单

好的教学方式等于成功的一半。

本书章节布局合理，全部采用模块化编写，可从头至尾，从易到难，由浅入深地学习，也可单独学习研究某一章节。

每一小节按“基础知识”、“例题讲解”、“思考练习”三部曲方式安排，完全符合科学有效的教练式教学模式。先精细化介绍基础知识，对所涉知识了然于胸，然后精选具有代表性的题目作为例题详细讲解，作为示范以例导学，最后安排大量练习题，采用一课多练的方式，检验并强化巩固所学知识。

(2) 活学活用，从入门到精通

“熟读唐诗三百首，不会作诗也会吟。”

本书打造了“一图一表”（一幅变量模型图，一张变量处理表）的特色例题讲解模式，生动形象，大大提高学习效率，入门从此变得轻松简单。同时提供海量针对性的习题，在不断练习的过程中逐渐学以致用、举一反三，俗话说“曲不离口，拳不离手”，熟才能生巧，只

有大量的强化练习方能真正将知识纳为己用，精通不再遥远。

(3) 现学现用，不懂也能用

“临阵磨枪，不快也光。”

本书亦可作为宏程序编程手册式书籍，章节模块化可单独查询学习，使用方便；每节例题采用模板式编写，可以“拿来主义”不求甚解直接套用。

君不见各种数控竞赛中宏程序大显身手，这是它高贵的一面；君不见它能像固定循环一样使编程和加工工作变得简单快捷，这是它普通的一面。驾驭它，你就能“高人一等”！毋庸置疑，宏程序是数控加工编程的高级内容，有人称它为“数控编程金字塔的塔尖”来形容它的高度与难度，通过本书的学习，它将高度犹存，难度不在。

还等什么？Let's go!

编著者

CONTENTS

目录



第1章 宏程序基础

Page 001

1.1 概述	1	1.6.1 概述	29
1.2 宏程序入门	4	1.6.2 简单宏程序调用 (G65)	29
1.3 变量	7	1.6.3 模态宏程序调用 (G66、G67)	32
1.3.1 概述	7	1.6.4 G指令宏程序调用	35
1.3.2 系统变量	10	1.6.5 M指令宏程序调用	37
1.4 算术和逻辑运算	17	1.6.6 M指令子程序调用	38
1.5 转移和循环语句	22		
1.6 宏程序的调用	29		

第2章 数控车削加工宏程序编程

Page 040

2.1 概述	40	零件编程模板	67
2.2 数控车削加工系列零件	40	2.4.2 工件原点在椭圆中心的正 椭圆类零件车削加工	71
2.3 数控车削加工固定循环	43	2.4.3 工件原点不在椭圆中心的正 椭圆类零件车削加工	77
2.3.1 外圆柱(锥)面加工循环	43	2.4.4 G65调用宏程序加工正椭圆 类零件车削加工	81
2.3.2 外圆柱(锥)螺纹加工 循环	46	2.4.5 倾斜椭圆类零件车削加工	83
2.3.3 梯形螺纹加工循环	48	2.4.6 抛物线类零件车削加工	88
2.3.4 圆弧螺纹加工循环	52	2.4.7 双曲线类零件车削加工	92
2.3.5 变螺距螺纹加工循环	55	2.4.8 正弦曲线类零件车削加工	94
2.3.6 钻孔加工循环	61	2.4.9 其他公式曲线类零件 车削加工	98
2.3.7 固定循环综合编程	63		
2.4 数控车削加工公式曲线类零件	67		
2.4.1 数控车削加工公式曲线类			

第3章 数控铣削加工宏程序编程

Page 102

3.1 概述	102	的铣削加工	103
3.2 数控铣削加工系列零件	103	3.2.2 相同轮廓的重复铣削加工	105
3.2.1 不同尺寸规格系列零件		3.3 数控铣削加工固定循环	107

3.4 零件平面铣削加工	109	3.6.4 大直径内螺纹铣削加工	148
3.4.1 长方形零件平面铣削加工	109	3.7 凹槽类零件铣削加工	152
3.4.2 圆形零件平面铣削加工	112	3.7.1 圆形凹槽类零件铣削加工 ...	152
3.5 公式曲线类零件铣削加工	116	3.7.2 矩形凹槽类零件铣削加工 ...	155
3.5.1 工件原点在椭圆中心的正椭圆类零件铣削加工	116	3.7.3 键槽类零件铣削加工	157
3.5.2 工件原点不在椭圆中心的正椭圆类零件铣削加工	119	3.7.4 阿基米德螺线凹槽类零件铣削加工	159
3.5.3 倾斜椭圆类零件铣削加工	122	3.7.5 空间曲线槽零件铣削加工 ...	162
3.5.4 抛物线类零件铣削加工	129	3.8 球面类零件铣削加工	164
3.5.5 双曲线类零件铣削加工	132	3.8.1 凸球面类零件铣削加工	164
3.5.6 其他公式曲线类零件铣削加工	135	3.8.2 凹球面类零件铣削加工	171
3.6 孔系类零件铣削加工	138	3.8.3 椭球面类零件铣削加工	174
3.6.1 直线点阵孔系铣削加工	138	3.9 凸台类零件铣削加工	178
3.6.2 圆周均分孔系铣削加工	140	3.9.1 圆锥台类零件铣削加工	178
3.6.3 矩形网式点阵孔系铣削加工	145	3.9.2 椭圆锥台类零件铣削加工 ...	182
		3.9.3 天圆地方凸台类零件铣削加工	185
		3.9.4 水平圆柱面铣削加工	189
		3.10 数控铣削加工零件轮廓倒角	193

第1章 宏程序基础

1.1 概述

(1) 用户宏程序的概念

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但是尺寸发生改变时，只能重新编程，灵活性和适用性差。另外在编制如椭圆等没有插补指令的公式曲线加工程序时，需要逐点算出曲线上的点，然后用直线或圆弧段逼近，如果零件表面粗糙度要求很高则需要计算很多点，程序庞大且不利于修改。利用数控系统提供的宏程序功能，当所要加工的零件形状不变只是尺寸发生了一定变化的情况时，只需要在程序中给要发生变化的尺寸加上几个变量和必要的计算公式，当加工的是椭圆等非圆曲线时，只需要在程序中利用数学关系来表达曲线，然后实际加工时，尺寸一旦发生变化，只要改变这几个变量的赋值参数就可以了。这种具有变量，并利用对变量的赋值和表达式来进行对程序编辑的程序叫宏程序。

数控系统供应商提供的宏程序称为系统宏程序，用户不能修改只能使用，如循环指令G70、G81等。客户自行编制的宏程序称为用户宏程序，可以修改、存储等。平常说的宏程序就是指用户宏程序。

宏程序可以较大地简化编程，扩展应用范围。宏程序适合图形类似只是尺寸不同的系列零件的编程，适合刀具轨迹相同只是位置参数不同的系列零件的编程，也适合抛物线、椭圆、双曲线等没有插补指令的曲线编程。

(2) 宏程序编程的基本特征

普通编程只能使用常量，常量之间不能运算，程序只能顺序执行，不能跳转。宏程序编程与普通程序编制相比有以下特征。

① **使用变量** 可以在宏程序中使用变量，使得程序更具有通用性，当同类零件的尺寸发生变化时，只需要更改宏程序主体中变量的值就可以了，而不需要重新编制程序。

② **可对变量赋值** 可以在宏程序调用命令中对变量进行赋值或在参数设置中对变量赋值，使用者只需要按照要求使用，而不必去理解整个宏程序内部的结构。

③ **变量间可进行演算** 在宏程序中进行变量的计算和算术逻辑运算，从而可以加工出非圆曲线轮廓和一些简单的曲面。

④ **程序运行可以跳转** 在宏程序中可以改变控制执行顺序。

(3) 宏程序与子程序的比较

① **相同之处** 宏程序是提高数控机床性能的一种特殊功能，使用中通常把能完成某一

功能的一系列指令像子程序一样存入存储器，然后用一个总指令代表它们，使用时只需给出这个总指令就能执行该功能。

子程序是将零件中常出现的几何形状完全相同的加工轨迹，编制成有固定顺序和重复模式的程序段，通常在几个程序中都会使用它。

宏程序的调用和子程序完全一样。

② **不同之处** 虽然子程序对编制相同加工操作的程序非常有用，但宏程序由于允许使用变量算术或逻辑运算及条件转移，使得编制相同甚至类似加工操作的程序更方便、更容易，如发展成打包好的自定义的固定循环。加工程序可利用一条简单的指令来调用宏程序，就像使用子程序一样，但是宏程序在调用指令中可对变量进行赋值。

(4) 宏程序的优点

① **长远性** 数控系统中随机携带有各种固定循环指令，这些指令是以宏程序为基础开发的通用的固定循环指令。通用循环指令有时对于工厂中实际的某一类特点的加工零件并不一定能满足加工要求，可以根据加工零件的具体特点，量身定制出适合这类零件特征的专用的宏程序，并固化在数控系统内部。这种专用的宏程序像使用普通固定循环指令一样调用，使数控系统增加了专用的固定循环指令，只要这一类零件继续生产，这种专用固定循环指令就一直存在并长期应用，因而，数控系统的功能得到增强和扩大。

② **共享性** 宏程序的编制确实存在相当的难度，要想编制出一个加工效率高、程序简洁、功能完善的宏程序更是难上加难，但是这并不影响宏程序的使用。正如设计一台电视机要涉及多方面的知识，考虑多方面的因素，是复杂的事情，但使用电视机却是一件相对简单的事情，使用者只要熟悉它的操作与使用，并不注重其内部构造和结构原理。宏程序的使用也是一样，使用者只需懂其功能、各参数的具体含义、使用限制注意事项即可，不必了解其设计过程、原理、具体程序内容。使用宏程序者不是必须要懂得宏程序，当然懂得宏程序可以更好地应用宏程序。

③ **多功能性** 宏程序的功能包含以下几个方面。

a. 相似系列零件的加工。同一类相同特征不同尺寸的零件，给定不同的参数，使用同一个宏程序就可以加工，编程得到大幅度简化。

b. 非圆曲线的拟合处理加工。对于椭圆、双曲线、抛物线、螺旋线、正（余）弦曲线等可以用数学公式描述的非圆曲线的加工，数控系统一般没有这样的插补功能，但是应用宏程序功能，可以将这样的非圆曲线用非常微小的直线段或圆弧段拟合加工，得到满足精度要求的非圆曲线。

c. 曲线交点的计算功能。在复杂零件结构中，许多节点的坐标是需要计算才能得到的，例如，直线与圆弧的交点、切点，直线与直线的交点，圆弧与圆弧的交点、切点等，不用人工计算并输入，只要输入已知的条件，节点坐标可以由宏程序计算完成并直接编程加工，在很大程度上增强了数控系统的计算功能，降低了编程的难度。

d. 人机界面及功能设计。数控系统一般针对通用机床开发，但也可用于专用机床，为适应专用机床的特点，数控系统显示的人机界面可用宏程序改变，但这需要厂家的技术支持。

④ **简练性与智能性** 宏程序是程序编制的高级阶段，程序编制的质量与编程人员的素质息息相关。高素质的编程人员在宏程序的编制过程中可以融入积累的工艺经验技巧，考虑轮廓要素之间的数学关系，应用适当的编程技巧，使程序非常简练，并且加工效果好。宏程序是由人工编制的，必然包含人的智能因素，程序中应考虑到各种因素对加工过程及精度的影响。

在质量上,自动编程产生的程序基本由 G00、G01、G02/G03 等简单指令组成,数据大部分是离散的小数点数据,难以分析、判别、查找错误,程序长度要比宏程序长几十倍甚至几百倍,不仅占用宝贵的存储空间,加工时间也要长得多。

(5) 编制宏程序的基础要求

宏程序的功能强大,但学会编制宏程序有相当的难度,要求掌握多方面的基础知识。

① **数学基础知识** 编制宏程序必须有良好的数学基础,数学知识的作用有多方面:计算轮廓节点坐标需要频繁的数学运算;在加工规律曲线、曲面时,必须熟悉其数学公式并根据公式编制相应的宏程序拟合加工,如椭圆的加工;更重要的是,良好的数学基础可以使人思维敏捷,具有条理性,这正是编制宏程序所需要的。

② **计算机编程基础知识** 宏程序是一类特殊的、实用性极强的专用计算机控制程序,其中许多基本概念、编程规则都是从通用计算机语言编程中移植过来的,所以学习 C 语言、BSAIC、FORTAN 等高级编程语言的知识,有助于快速理解并掌握宏程序。

③ **一定的英语基础** 在宏程序编制过程中需要用到许多英文单词或单词的缩写,掌握一定的英语基础可以正确理解其含义,增强分析程序和编制程序的能力;再者,数控系统面板按键及显示屏幕中也有为数不少的英语单词,良好的英语基础有利于熟练操作数控系统。

④ **耐心与毅力** 相对于普通程序,宏程序显得枯燥而难懂。编制宏程序过程需要灵活的逻辑思维能力,调试宏程序需要付出更多的努力,发现并修正其中的错误需要耐心与细致,更要有毅力从一次次失败中汲取经验教训并最终取得成功。

(6) FANUC 用户宏程序的分类

FANUC 用户宏程序功能分 A、B 两类,在功能上差异并不大,但在编程和分析判读方面,B 类宏程序要比 A 类宏程序清晰容易得多。A 类宏程序使用的是 G65 Hm 指令,指令格式长而含义表达含糊,B 类宏程序使用的是高级语言编程,表达式简单明了而且含义清晰。

例如,要表达 $\#101 = \#102 + \#103$;

A 类宏程序为 G65 H02 P#101 Q#102 R#103; (“G65 H02”代表加法运算)

B 类宏程序为 $\#101 = \#102 + \#103$;

在用户宏程序发展的初期,A 类宏程序用于车床数控系统(如 FANUC 0TD)较多,B 类宏程序用于铣床和加工中心数控系统较多,而现在绝大部分 FANUC 系统(如 FANUC 0i)中都应用了 B 类宏程序,使宏程序的编制得到了简化,本书中均以 FANUC 0i 系统为蓝本的 B 类宏程序进行讲解。

(7) 用户宏程序的使用

宏程序语句(简称宏指令)既可以在主程序体中使用,也可以当作子程序来调用。

① 放在主程序体中

```
...  
N50 #100=30.0;  
N60 #101=20.0;  
N70 G01 X#100 Y#101 F500;  
...
```

② 当作子程序调用

主程序:

```
O1001;                                (主程序号)  
...
```

```

...
N100 G65 P1002 L2 A1.0 B2.0;    (调用 1002 号宏程序 2 次, 将值传递给 #1 和 #2 变量)
...
...
M30;                               (主程序结束)

```

宏程序:

```

O1002;                             (宏程序号)
N10 #3=#1+#2;                       (赋值语句)
N20 IF [#3GT360]GOTO40;             (条件转移语句)
N30 G00 G91 X#3;                    (快速点定位)
N40 M99;                             (宏程序结束并返回主程序)

```



思考练习

1. FANUC 用户宏程序有哪些种类?
2. 宏程序编程与普通程序编制相比有哪四大特征?
3. 用户宏程序有哪些优点? 它可以用于哪些方面的数控编程?
4. 用户宏程序的用法有哪几种?

1.2 宏程序入门

为了让读者对宏程序有一个比较简单的认识, 这里先介绍两个宏程序入门例题, 它们分别属于将宏指令放在主程序体中和当作子程序来调用的两种不同应用方式。

【例 1-1】 数控铣削精加工图 1-1 (a) 所示矩形外轮廓, 要求采用宏程序指令编制加工程序。

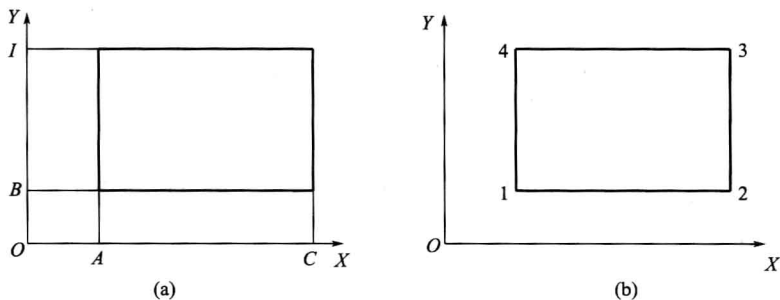


图 1-1 矩形外轮廓数控铣削加工

解 假定起刀点在 O 点, 如图 1-1(b) 所示, 按 $O \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow O$ 的走刀轨迹加工 (不考虑刀具补偿等问题), 则加工程序如下:

```

N10 G00 XA YB;                     (从 O 点快速点定位至 1 点)
N20 G01 XC F100;                   (直线插补至 2 点)
N30 YI;                             (直线插补至 3 点)
N40 XA;                             (直线插补至 4 点)
N50 YB;                             (直线插补至 1 点)

```

N60 G00 X0 Y0; (返回 O 点)

将程序中变量 A、B、C、I 用宏程序中的变量 #i 来代替, 设字母与的 #i 的对应关系为 (即将 A、B、C、I 分别赋值给 #1、#2、#3 和 #4):

#1=A
#2=B
#3=C
#4=I

则编制宏程序如下:

```
N10 #1=A; (将 A 值赋给 #1)
N20 #2=B; (将 B 值赋给 #2)
N30 #3=C; (将 C 值赋给 #3)
N40 #4=I; (将 I 值赋给 #4)
N50 G00 X#1 Y#2; (从 O 点快速点定位至 1 点)
N60 G01 X#3 F100; (直线插补至 2 点)
N70 Y#4; (直线插补至 3 点)
N80 X#1; (直线插补至 4 点)
N90 Y#2; (直线插补至 1 点)
N100 G00 X0 Y0; (返回 O 点)
```

当加工同一类尺寸不同的零件时, 只需改变宏指令的数值即可, 而不必针对每一个零件都编一个程序。当然, 实际使用时一般还需要在上述程序中加上坐标系设定、刀具半径补偿和 F、S、T 等指令。

【例 1-2】 调用宏程序车削加工图 1-2 所示的台阶轴零件。

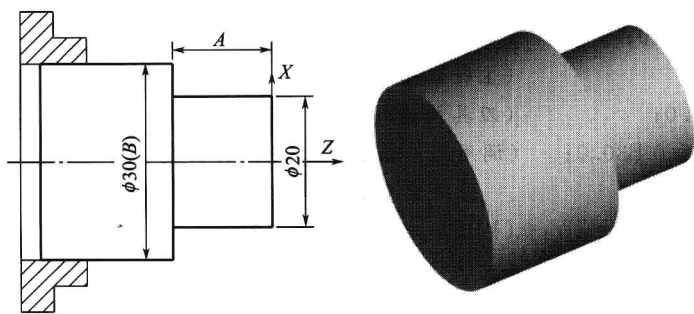


图 1-2 台阶轴零件数控车削加工

解 图中标注 A 的轴肩通常有不同长度, 采用宏程序编程可以满足加工不同 A 尺寸工件的需要。为了加工该工件, 需要按照一般的格式编制主程序, 在主程序中通常是刀具到达准备开始加工位置时, 有一程序段调用宏程序, 宏程序执行结束, 则返回主程序中继续执行。编制加工程序如下。

主程序:

```
O1010; (主程序号)
N10 G50 X150.0 Z50.0; (工件坐标系设定)
N20 S550 M03; (主轴正转)
```




```

N30 G00 X20.0 Z2.0;      (刀具快速到达切削起始点)
N40 G65 P1011 A15;        (调用 1011 号用户宏程序, 将轴肩长度“15”赋值给变量#1)
N50 G01 X30.0;           (车削轴肩)
N60 G00 X150.0 Z200.0;   (快速返回刀具起始点)
N70 M05;                 (主轴停转)
N80 M30;                 (程序结束)

```

宏程序:

```

O1011;                   (宏程序号)
G01 Z—#1 F0.2;          (车削外圆, 可获得任意轴肩长度)
M99;                     (返回主程序)

```

在主程序中, N40 程序段用 G65 指令调用 1011 号宏程序, A15 表示将轴肩的长度 15mm 赋值给变量 #1。车削轴端外圆并保证所需长度尺寸是通过宏程序中下面程序段实现的:

```
G01 Z—#1 F0.2;
```

如果用一般程序加工轴肩长度为“15”的外圆, 可输入下面的程序段:

```
G01 Z—15.0 F0.2;
```

然而, 这只能加工这种长度的工件。宏程序允许用户加工任意所需长度的工件, 这可以通过改变 G65 指令中地址 A 后的数值实现。

轴肩的长度加工完成后, 执行 M99 返回到主程序, 加工轴肩端面并获得所需直径。如果轴肩直径 (图 1-2 中 B 尺寸) 也需要变化, 也可以通过宏程序实现。为此, 在主程序中还需要加入地址 B, 程序可修改如下。

主程序:

```

O1012;                   (主程序号)
N10 G50 X150.0 Z50.0;    (工件坐标系设定)
N20 S550 M03;            (主轴正转)
N30 G00 X20.0 Z2.0;      (刀具快速到达切削起始点)
N40 G65 P1013 A15.0 B30.0; (调用用户宏程序, 将轴肩长度“15”和轴肩直径“30”分别赋值给变量#1和#2)
N50 G00 X150.0 Z200.0;   (快速返回刀具起始点)
N60 M05;                 (主轴停转)
N70 M30;                 (程序结束)

```

宏程序:

```

O1013;                   (宏程序号)
G01 Z—#1 F0.2;          (车削外圆, 可获得任意轴肩长度)
X#2;                     (车削轴肩可得到任意直径)
M99;                     (返回主程序)

```

该程序中通过地址 B 把直径“30”赋值给变量 #2, 只需要修改 N40 程序段中的 A、B 值即可加工不同轴肩长度直径的工件。

从以上例子可以看出, 用户宏程序中可以用变量代替具体数值, 因而在加工同一类型工件时, 只需对变量赋不同的值, 而不必对每一个零件都编一个程序。



思考练习

1. 编制精加工图 1-3(a) 所示直径为 D 、长度为 L 的外圆柱面的部分程序段，请完成填空并回答问题。

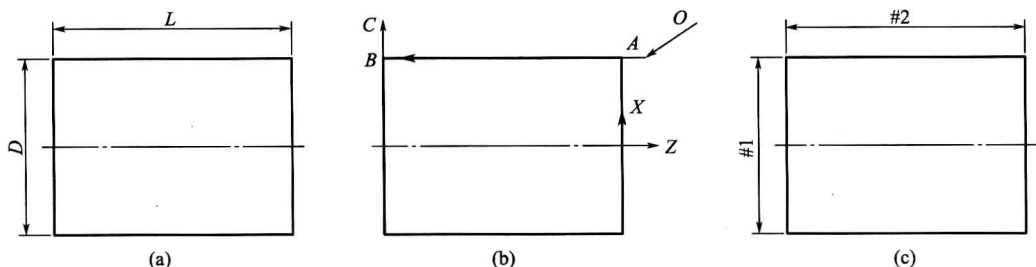


图 1-3 圆柱面的加工

如图 1-3(b) 所示，建立工件坐标系原点于右端面与轴线的交点上，按 $O \rightarrow A \rightarrow B \rightarrow C$ 的路线加工（设 A 点距离工件右端面 2mm， C 点距离工件外圆面 5mm），编制部分加工程序如下。

```
O→A: G00 X[D] Z2.0;
A→B: G01 X[D] Z[-L] F0.1;
B→C: G00 X[D+10] Z _____;
```

如图 1-3(c) 所示，若用 #1 替代直径 D ，用 #2 替代长度 L ，则将上述程序修改为：

```
O→A: G00 X#1 Z2.0;
A→B: G01 X _____ Z-#2 F0.1;
B→C: G00 X[#1+10] Z-#2;
```

若要求精加工“ $\phi 40 \times 25$ ”的外圆柱面，执行如下程序即可：

```
#1=40;
#2=25;
G00 X#1 Z2.0;
G01 X#1 Z-#2 F0.1;
G00 X[#1+10] Z-#2;
```

请问若要求精加工“ $\phi 32 \times 20$ ”的外圆柱面，程序应如何修改？

2. 试简述采用常量编程（普通程序）和用变量编程（宏程序）各有何特点。

1.3 变量

1.3.1 概述

值不发生改变的量称为常量，如“ $G01 X100 Y200 F300$ ”程序段中的“100”、“200”、“300”就是常量，而值可变的量称为变量，在宏程序中使用变量来代替地址后面的具体数值，如“ $G01 X\#4 Y\#5 F\#6$ ”程序段中的“#4”、“#5”、“#6”就是变量。变量可以在程序中或 MDI 方式下对其进行赋值。变量的使用可以使宏程序具有通用性，并且在宏程序中可以使用多个变量，彼此之间用变量号码进行识别。

(1) 变量的表示形式

变量的表示形式为： $\#i$ ，其中，“#”为变量符号，“ i ”为变量号，变量号可用 1、2、

3 等数字表示,也可以用表达式来指定变量号,但其表达式必须全部写入方括号“[]”中,例如#1和#[#1+#2+10]均表示变量,当变量#1=10,变量#2=100时,变量#[#1+#2+10]表示#120。

表达式是指用方括号“[]”括起来的变量与运算符的结果。表达式有算术表达式和条件表达式两种。算术表达式是使用变量、算术运算符或者函数来确定的一个数值,如[10+20]、[#10*#30]、[#10+42]和[1+SIN[30]]都是算术表达式,它们的结果均为一个具体的数值。条件表达式的结果是零(假)(FALSE)或者任何非零值(真)(TRUE),如[10GT5]表示一个“10大于5”的条件表达式,其结果为真。

(2) 变量的类型

根据变量号,变量可分成四种类型,如表1-1所示。

表 1-1 变量类型

变量号	变量类型	功 能
#0	空变量	该变量总是空的,不能被赋值(只读)
#1~#33	局部变量	局部变量只能在宏程序内部使用,用于保存数据,如运算结果等。当电源关闭时,局部变量被清空,而当宏程序被调用时,(调用)参数被赋值给局部变量
#100~#149(#199) #500~#531(#999)	公共变量	公共变量在不同宏程序中的意义相同。当电源关闭时,变量#100~#149被清空,而变量#500~#531的数据仍保留
#1000~#9999	系统变量	系统变量可读、可写,用于保存 NC 的各种数据项,如:当前位置、刀具补偿值、机床模态等

注意:

① 公共变量#150~#199, #532~#999是选用变量,应根据实际系统使用。

② 局部变量和公共变量称为用户变量。局部变量和公共变量可以有“0”值或在下述范围内的值: $-10^{47} \sim 10^{-29}$ 或 $10^{-29} \sim 10^{47}$,如计算结果无效(超出取值范围)时,发出编号“111”的错误警报。

(3) 变量的引用

将跟随在地址符后的数值用变量来代替的过程称为引用变量。同样,引用变量也可以采用表达式。在程序中引用(使用)变量时,其格式为在指令字地址后面跟变量号。当用表达式表示变量时,表达式应包含在一对方括号内,如:“G01 X[#1+#2] F#3;”。

变量引用的注意事项:

① 被引用变量的值会根据指令地址的最小输入单位自动进行四舍五入,例如,程序段“G00 X#1”,给变量#1赋值12.3456,在1/1000mm的CNC上执行时,程序段实际解释为“G00 X12.346”。

② 要使被引用的变量值反号,在“#”前加前缀“-”即可,如“G00 X-#1;”。

③ 当引用未定义(赋值)的变量时,这样的变量称为“空”变量(变量“#0”总是空变量),该变量前的指令地址被忽略,如:#1=0, #2=“空”(未赋值),执行程序段“G00 X#1 Y#2;”,结果为“G00 X0”。

④ 当引用一个未定义的变量时,地址本身也被忽略。

⑤ 变量引用有限制,变量不能用于程序号“O”、程序段号“N”、任选段跳跃号“/”,例如,如下变量使用形式均是错误的。

```
O#1;
/#2 G00 X100.0;
N#3 Y200.0;
```

(4) 变量的赋值

赋值是指将一个数赋予一个变量。变量的赋值方式有两种。

① **直接赋值** 变量可以在操作面板上用 MDI 方式直接赋值,也可在程序中以等式方式赋值,但等号左边不能用表达式。

例如,“ $\#1=100;$ ”中,“ $\#1$ ”表示变量;“ $=$ ”表示赋值符号,起语句定义作用;“100”就是给变量 $\#1$ 赋的值。

“ $\#100=30+20;$ ”中,将表达式“ $30+20$ ”赋值给变量 $\#100$,即 $\#100=50$ 。

直接赋值相关注意事项:

a. 赋值符号(=)两边内容不能随意互换,左边只能是变量,右边可以是数值、表达式或者变量。

b. 一个赋值语句只能给一个变量赋值。

c. 可以多次给一个变量赋值,但新的变量值将取代旧的变量值,即最后赋的值有效。

d. 在程序中给变量赋值时,可省略小数点。例如,当 $\#1=123$ 被定义时,变量 $\#1$ 的实际值为123.0。

e. 赋值语句在其形式为“变量=表达式”时具有运算功能。在运算中,表达式可以是数值之间的四则运算,也可以是变量自身与其他数据的运算结果,如:“ $\#1=\#1+1$ ”,则表示新的 $\#1$ 等于原来的 $\#1+1$,这点与数学等式是不同的。

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,但是如果对计算机编程语言(例如C语言)有一定了解的话,对此应该更易理解。

f. 赋值表达式的运算顺序与数学运算的顺序相同。

② **自变量赋值** 宏程序以子程序方式出现时,所用的变量可在宏程序调用时赋值。例如,程序段“G65 P1020 X100.0 Y30.0 Z20.0 F100”,该处的X、Y、Z不代表坐标字,F也不代表进给字,而是对应于宏程序中的局部变量号,变量的具体数值由自变量后的数值决定(详见1.6“宏程序的调用”)。



思考练习

一、选择题

- 宏程序中的 $\#110$ 属于()。
A) 常量 B) 局部变量 C) 系统变量 D) 公共变量
- 若 $\#103=50$ 时, $F\#103$ 表示()。
A) F50 B) F103 C) F0 D) 表示形式有误
- 若 $\#110=100$ 时,对于 $Z-\#110$ 最准确的表示是()。
A) $Z-100.0$ B) $Z-100$ C) $Z100.0$ D) $Z-110.0$
- 若 $\#120=3$ 时, $G\#120$ 表示()。
A) G03 B) G120 C) G00 D) 表示形式有误
- 若 $\#130=90$ 时, $N\#130$ 表示()。
A) N90 B) N130 C) N0 D) 表示形式有误
- 执行如下两程序段后,N2程序段计算的是变量()的值,其值为()。
N1 $\#1=3;$