轻松掌握

华中宏程序

——编程技巧与实例精解

杜 军 龚水平 编著

QINGSONG ZHANGWO HUAZHONG HONGCHENGXU
BIANCHENG JIQIAO YU SHILI JINGJIE

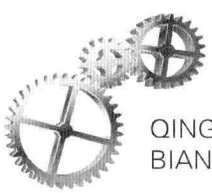
化学工业出版社

轻松掌握

华中宏程序

——编程技巧与实例精解

杜 军　龚水平　编著



QINGSONG ZHANGWO HUAZHONG 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华中数控宏程序编程的书。

本书是实用性非常强的数控技术用书，详细介绍了华中数控系统宏程序的基础知识、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，采用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 70 余道典型例题详细讲解以期重难点突破，最后精心设计了 180 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的人员使用，也可供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

轻松掌握华中宏程序——编程技巧与实例精解 / 杜军，龚水平编著. —北京：化学工业出版社，2012.2

ISBN 978-7-122-13260-4

Ⅰ. 轻… Ⅱ. ①杜… ②龚… Ⅲ. 数控机床-数控系统 Ⅳ. TG659

中国版本图书馆 CIP 数据核字（2012）第 004188 号

责任编辑：张兴辉　　文字编辑：陈　喆

责任校对：洪雅姝　　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：三河市延风印装厂

787mm×1092mm　1/16　印张 18¼　字数 450 千字　　2012 年 4 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：49.00 元

前言 FOREWORD

这是一本让你轻松实现从入门到精通华中数控宏程序编程的书。

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：某光轴零件，其尺寸在图 1 中用“ϕ30×40”表示，这是常量形式，如图 2 所示是用符号“$D\times L$”表示，图 3 则是用数控机床认识的符号“#1×#2”表示，不难理解，其实仅是用不同的表示符号进行了置换而已。

设工件坐标原点在工件右端面与轴线的交点上，数控车削精加工路线为 $A\rightarrow B$，可编制加工程序分别为“G01 X30 Z-40”、“G01 X[D] Z[-L]”和“G01 X[#1] Z[-#2]”，其中“G01X[#1] Z[-#2]”就是一个宏程序语句，符号“#1”、“#2”就是变量！

定义有了：含有“#i”（变量）符号的程序就叫宏程序！很简单的！

比较一下，“G01 X30 Z-40”只能加工出“ϕ30×40”的圆柱面；“G01 X[D] Z[-L]”数控机床无法识别；而“G01 X[#1] Z[-#2]”，若令#1=30、#2=40，可加工出“ϕ30×40”的圆柱面，若令#1=12、#2=10，则可加工出“ϕ12×10”的圆柱面……仅改变变量的值即可加工出不同直径和长度的外圆柱面！

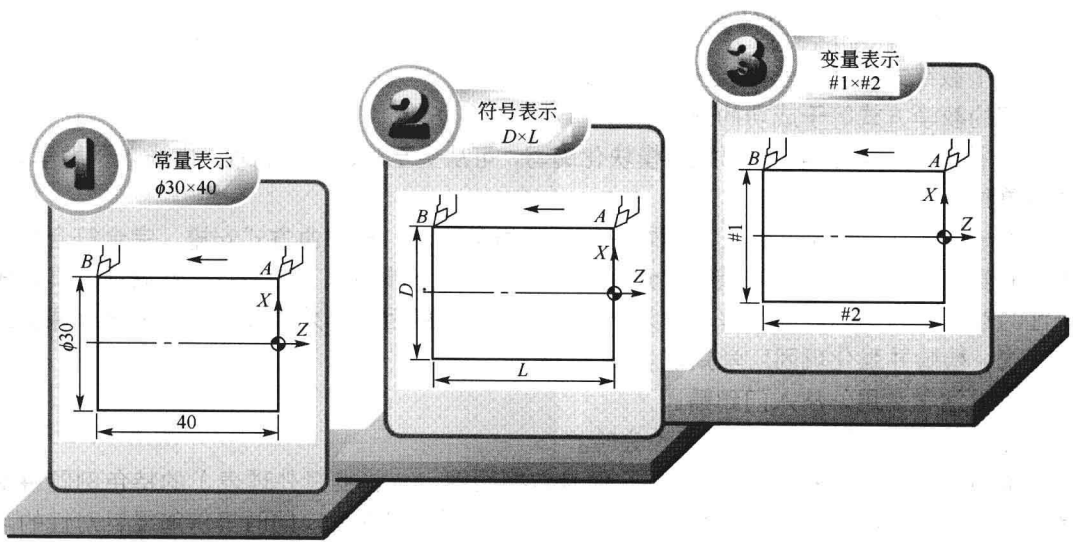


图 1　　图 2　　图 3

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，还可设为专用 G 指令方便调用。

（2）实现曲线插补

一般数控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，可大大拓展数控系统插补指令。

关于宏程序能给你带来的改变，也许从下图能有所了解。

“简易”

使用简便：一条指令调用宏程序即可完成期望动作。
内容简练：与自动编程相比，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、判别和查找错误，程序长度要比宏程序长几十倍甚至几百倍，占用宝贵的存储空间不说，加工时间也要长得多。

“变易”

执行“#1=20”程序段之后变量“#1”的值为“20”，如果执行“#1=#1-2”程序段之后，“#1”的值是“18”，接着执行程序“IF#1GT5；#1=72；ENDIF”后变量“#1”的值又会变成“72”。这就是宏程序的神奇之处：变量可万变！

“学易”

正如我们熟知的《易经》所讲：有变必有不变。
虽然宏程序编程思路和方法不尽相同，但事实上还是有一定的规律可循：如本书分析有变量处理表，编程提供编程模板，即使变量万变，宏程序内容也无需改变，正所谓“万变不离其宗”。

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”（一幅几何参数模型图、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，只有大量地强化练习方能真正将知识纳为己用，精通则不再遥远。

（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工具书，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”，以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

目录

第 1 章　宏程序基础　Page 1

第 2 章　数控车削加工宏程序编程　Page 25

第 3 章　数控铣削加工宏程序编程　Page 78

参考答案

Page 182

附录

Page 252

参考文献

Page 282

第1章 宏程序基础

1.1 概述

（1）用户宏程序概念

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，比如循环指令 G71、G81 等。用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　可以在宏程序调用指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　在宏程序中可以进行变量的四则运算和算术逻辑运算，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）宏子程序与子程序的比较

① 相同之处　类似于子程序存储后在主程序中调用的宏程序称为宏子程序。宏子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和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。

宏子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，但宏子程序由于允许使用变量进行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子程序，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实际加工中某一类特点的加工零件并不一定能满足加工要求，对此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，因此，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，并不需要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、各参数的具体含义和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宏程序，当然懂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　宏程序是数控加工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，使程序非常精练，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，自动编程生成的加工程序基本由 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、判别和查找错误，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，要求有多方面的基础知识。

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的思维敏捷，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。编制宏程序过程中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）用户宏程序的使用

宏程序语句（简称宏指令）既可以在主程序中使用，也可以当作子程序来调用。

① 放在主程序体中：

```
…
#100=30
#101=20
G01 X[#100] Y[#101] F150
…
```

② 当作子程序调用：

主程序：

```
%1001                          （主程序号）
…
…
M98 P1002 A10 B22              （调用 1002 号宏程序，将值传递给变量#0 和#1）
…
…
M30                            （主程序结束）
```

宏子程序：

```
%1002                          （宏程序号）
#3=#1+#2                       （赋值语句）
IF#3GT360                      （条件判断语句）
```

```
#3=#1-#2                    (重新赋值)
ENDIF                       (条件结束)
G00 G91 X[#3]               (快速点定位)
M99                         (宏子程序结束并返回主程序)
```

思考练习

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

1.2 宏程序入门

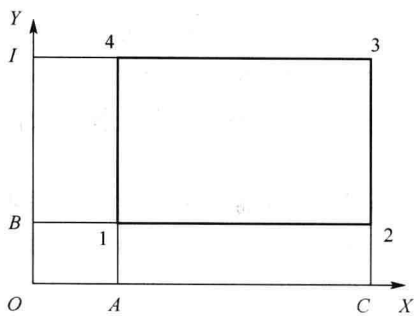


图 1-1 矩形外轮廓数控铣削加工

为了让读者对宏程序有一个比较简单的认识，本节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中和当作子程序来调用两种不同的应用方式。

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

解：假定起刀点在 O 点，如图 1-1 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），则部分加工程序如下：

```
N10 G00 X[A] Y[B]           (从O点快速点定位至1点)
N20 G01 X[C] F100           (直线插补至2点)
N30 Y[I]                    (直线插补至3点)
N40 X[A]                    (直线插补至4点)
N50 Y[B]                    (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#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】 调用宏程序完成数控车削加工如图 1-2 所示的台阶轴零件。

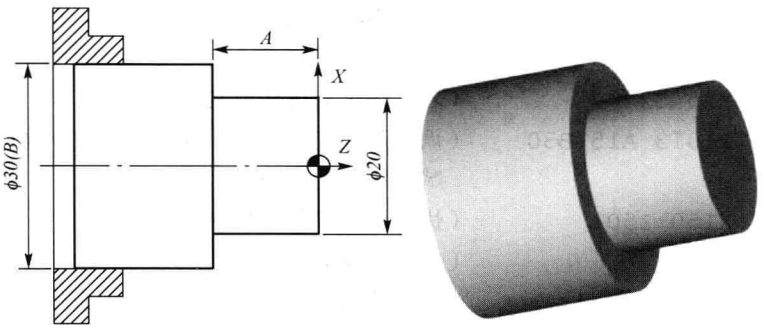


图 1-2　台阶轴零件数控车削加工

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。加工该零件时，主程序仍然按照普通格式编制，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子程序，宏子程序执行结束后返回主程序中继续执行。编制加工程序如下。

主程序：

```
%1010                        （主程序号）
N10 T0101 S550 M03 F120      （加工参数设定）
N20 G00 X150 Z50             （刀具进给到起刀点）
N30 G00 X20 Z2               （刀具快速到达切削起始点）
N40 M98 P1011 A15            （调用 1011 号用户宏程序，将轴肩长度 15 赋值给变量#0）
N50 G01 X30                  （车削轴肩）
N60 G00 X150 Z50             （快速返回刀具起始点）
N70 M05                      （主轴停转）
N80 M30                      （程序结束）
```

宏子程序：

```
%1011                        （宏子程序号）
G01 Z[-#0]                   （车削外圆，可获得任意轴肩长度）
M99                          （子程序结束并返回主程序）
```

在主程序中，N40 程序段用 M98 指令调用 1011 号宏子程序，A15 表示将轴肩的长度 15mm 赋值给变量#0。车削轴端外圆并保证所需长度尺寸是通过宏子程序中下面程序段实现的：

```
G01 Z[-#0]
```

如果用一般程序加工轴肩长度为 15 的外圆，输入的是下面的程序段：

```
G01 Z-15
```

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，可以通过改变 M98 指令中地址 A 后的数值来实现。

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：

```
%1012                          (主程序号)
N10 T0101 S550 M03 F120        (加工参数设定)
N20 G00 X150 Z50               (刀具快进到起刀点)
N30 G00 X20 Z2                 (刀具快速到达切削起始点)
N40 G65 P1013 A15 B30          (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#0 和#1)
N50 G00 X150 Z50               (快速返回刀具起始点)
N60 M05                        (主轴停转)
N70 M30                        (程序结束)
```

宏程序：

```
%1013                          (宏程序号)
G01 Z[-#0]                     (车削外圆，可获得任意轴肩长度)
X[#1]                          (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 把直径值 30mm 赋值给变量#1，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* 的外圆柱面部分加工程序段，请完成填空并回答问题。

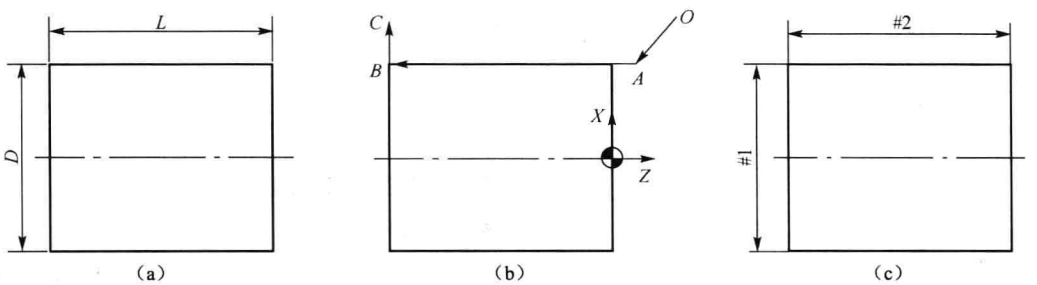


图 1-3　圆柱面加工

如图 1-3（b）所示，建立工件坐标原点于工件右端面与轴线的交点上，按 *O*→*A*→*B*→*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

A→*B*: G01 X[D] Z[-L]

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

A→*B*: G01 X__Z[-#2]

B→*C*: G00 X[#1+10] Z[-#2]

若要求精加工 ϕ40×25 的外圆柱面，执行如下程序即可：

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

请问若要求精加工 ϕ32×20 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）变量的定义

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（常数），而其数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。

（2）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 1～4 位数字表示，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*PI/180]]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。

（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。无论是用数字还是用表达式表示变量号时，变量均应包含在一对方括号内，如：“G01 X[#1+#2] F[#3]”。

变量可以用来代替程序中的数据，如尺寸、刀补号、G 指令编号等。变量的使用，给程序的设计带来了极大的灵活性。

使用变量前，变量必须带有正确的值。如：

```
#1=25              (#1 的初始值为 25)
G01 X[#1]          (表示 G01 X25)
#1=-10             (运行过程中可以随时改变#1 的值)
G01 X[#1]          (表示 G01 X-10)
```

用变量不仅可以表示坐标，还可以表示 G、M、F、D、H、M、X、Y 等各种代码后的数字。如：

```
#2=3               (#2 赋值)
G[#2] X30          (表示 G03 X30)
```

变量引用的注意事项：

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

② 当引用未定义（赋值）的变量时，这样的变量称为“空”变量，其值为“0”，如：#1=0，#2=“空”（未赋值），执行程序段“G00 X[#1] Y[#2]”的结果为“G00 X0 Y0”。

③ 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
N[#3] Y200
```

（4）变量的赋值

把一个常数或表达式的值赋给一个宏变量称为赋值。变量的赋值方式有直接赋值和自变量赋值两种。

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。变量的直接赋值种类及格式如表 1-1 所示。

表 1-1　变量的直接赋值种类及格式

种　类	格　式
常量赋值	#i=（具体数值）
变量赋值	#i=#j 或#i=（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．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”表示新的#1 的值等于原来的#1 的值加 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。

e．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“M98 P1020 X100 Y30 Z20 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。

（5）变量的类型

① 局部变量　#0～#49 变量是局部变量。局部变量的作用范围是当前程序（在同一个程序号内），如果在主程序或不同子程序里出现了相同名称（变量号）的变量，它们不会相互干扰，值也可以不同。

例如：

```
%100          （主程序号）
N10 #3=30     （主程序中#3 为 30）
M98 P101      （进入子程序后#3 不受影响）
#4=#3         （#3 仍为 30，所以#4=30）
M30           （主程序结束）
%101          （宏子程序号）
#4=#3         （这里的#3 不是主程序中的#3，由于#3 在子程序内没有定义，所以
                #3=0，则#4=0）
#3=18         （这里使#3 的值为 18，不会影响主程序中的#3）
M99           （子程序结束）
```

② 全局变量　编号#50～#199 的变量是全局变量（注：其中#100～#199 也是刀补变量）。全局变量的作用范围是整个零件程序，不管是主程序还是子程序，只要名称（变量号）相同就是同一个变量，带有相同的值，在某个地方修改它的值，所有其他地方都受影响。

例如：

```
%100          （主程序号）
#50=30        （先使#50 的值为 30）
M98 P101      （进入子程序）
#4=#50        （#50 变为 18，所以#4=18）
M30           （程序结束）
%101          （宏子程序号）
#4=#50        （#50 的值在子程序里也有效，所以#4=30）
#50=18        （这里使#50=18，然后返回）
M99           （子程序结束）
```

如果只有全局变量，由于变量名不能重复，就可能造成变量名不够用；全局变量在任何地方都可以改变它的值，这是它的优点，也是它的缺点。说是优点，是因为参数传递很方便；说是缺点，是因为当一个程序较复杂的时候，一不小心就可能在某个地方用了相同的变量名或者改变了它的值，造成程序混乱。局部变量的使用，解决了同名变量冲突的问题，编写子程序时，不需要考虑其他地方是否用过某个变量名。

在一般情况下，应优先考虑选用局部变量。局部变量在不同的子程序里，可以重复使用，不会互相干扰。如果一个数据在主程序和子程序里都要用到，就要考虑用全局变量。