



综合铣削编程训练

一、子程序调用

二、镜像、旋转、缩放指令

三、宏编程指令（第三篇介绍）

一、子程序调用技术

- 1、子程序的含义
- 2、主、子程序结构的异同
- 3、调用子程序指令格式
- 4、主-子程序调用关系
- 5、主-子程序结构书写
- 6、主-子程序结构应用关键
- 7、编程举例

1、子程序的含义

➤什么是子程序？

在编制加工程序中，有时会出现有规律、重复出现的程序段。

将程序中重复的程序段单独抽出，并按一定格式单独命名，称之为子程序。

子程序的含义

- 采用子程序的意义
 - 使复杂程序结构明晰
 - 程序简短
 - 增强数控系统编程功能

2、主、子程序结构异同

相同:

都是完整的程序。包括程序号、程序段、程序结束指令

不同:

•程序结束指令不同

主程序: M02 或 M30

子程序: M99

•子程序不能单独运行, 由主程序或上层子程序调用执行。

3、调用子程序的指令格式

➤子程序调用的指令格式:

M98 P 单次调用指令，P后跟被调用的子程序号

M98 P L 重复调用子程序指令，L后跟重复调用的次数

➤子程序的格式:

•子程序号：是调用入口地址，必须和主程序中的子程序调用指令中所指向的程序号一致。

•子程序结束： M99

4、主子程序调用关系



❖ 逐层调用，逐层返回。

❖ 可实现八层嵌套

5、主-子程序结构书写

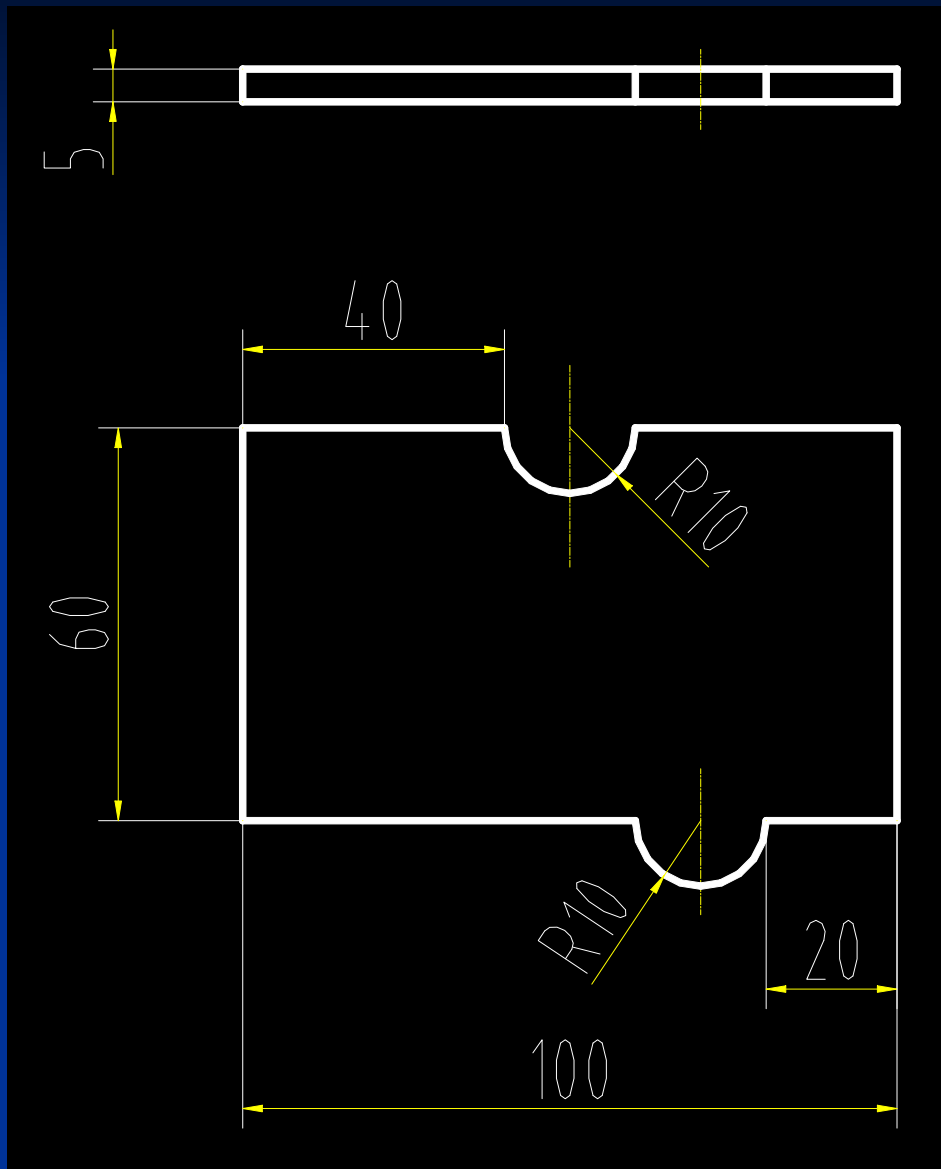
主—子程序：

- 写在一个文件中
- 主程序写在前
- 子程序写在后
- 两者之间空几行作分隔

6、主-子程序结构应用关键

- ❖ 找出重复程序段规律，确定子程序。
- ❖ 将要变化的部分写在主程序，不变的部分作子程序。
- ❖ 主—子程序接口：保证主程序调用和子程序返回正确的衔接
如：从某点进入子程序，返回时也固定在该点。

7、子程序调用编程举例（一）

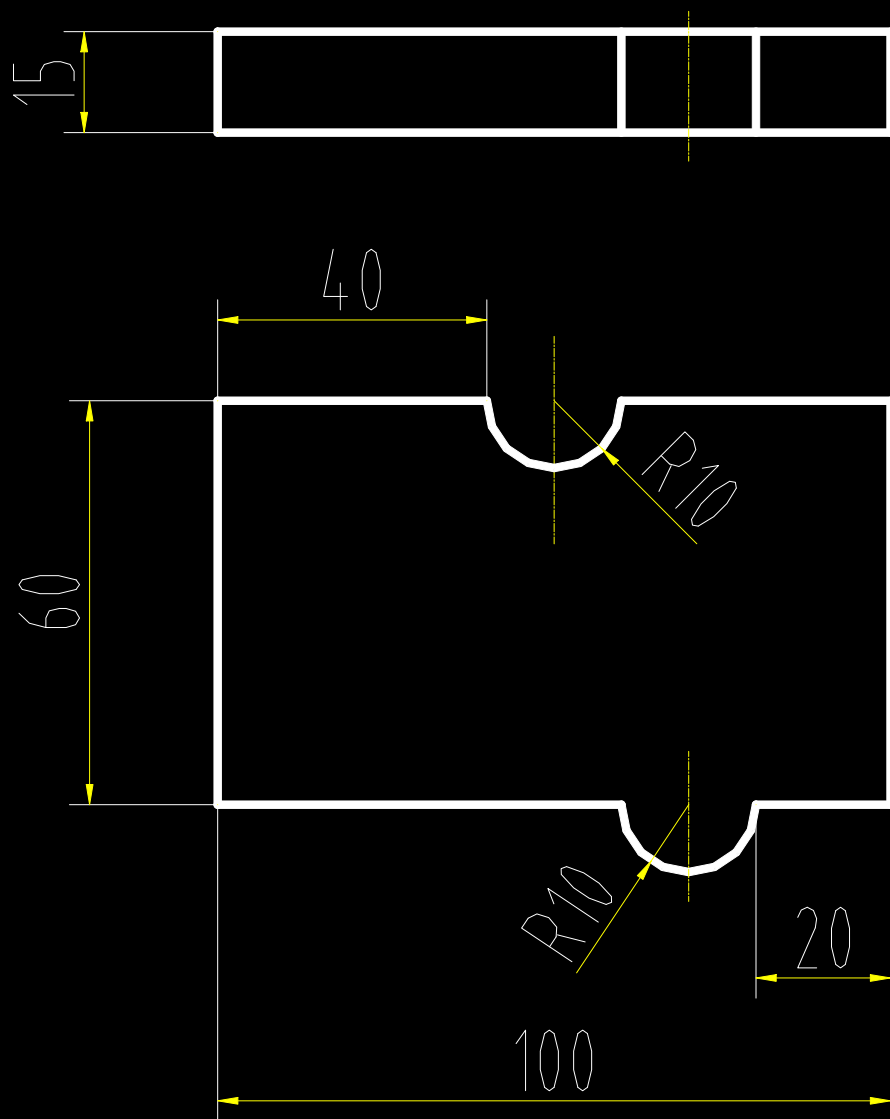


(1) 利用子程序调用实现外轮廓粗、精加工。

如图所示零件

- ✓ 用 $\phi 8$ 立铣刀
- ✓ 采用粗、精铣两刀完成外轮廓加工。
- ✓ 精加工余量 0.25mm
- ✓ 深度方向一次下刀完成。
(暂不考虑装夹)

7、子程序调用编程举例(二)

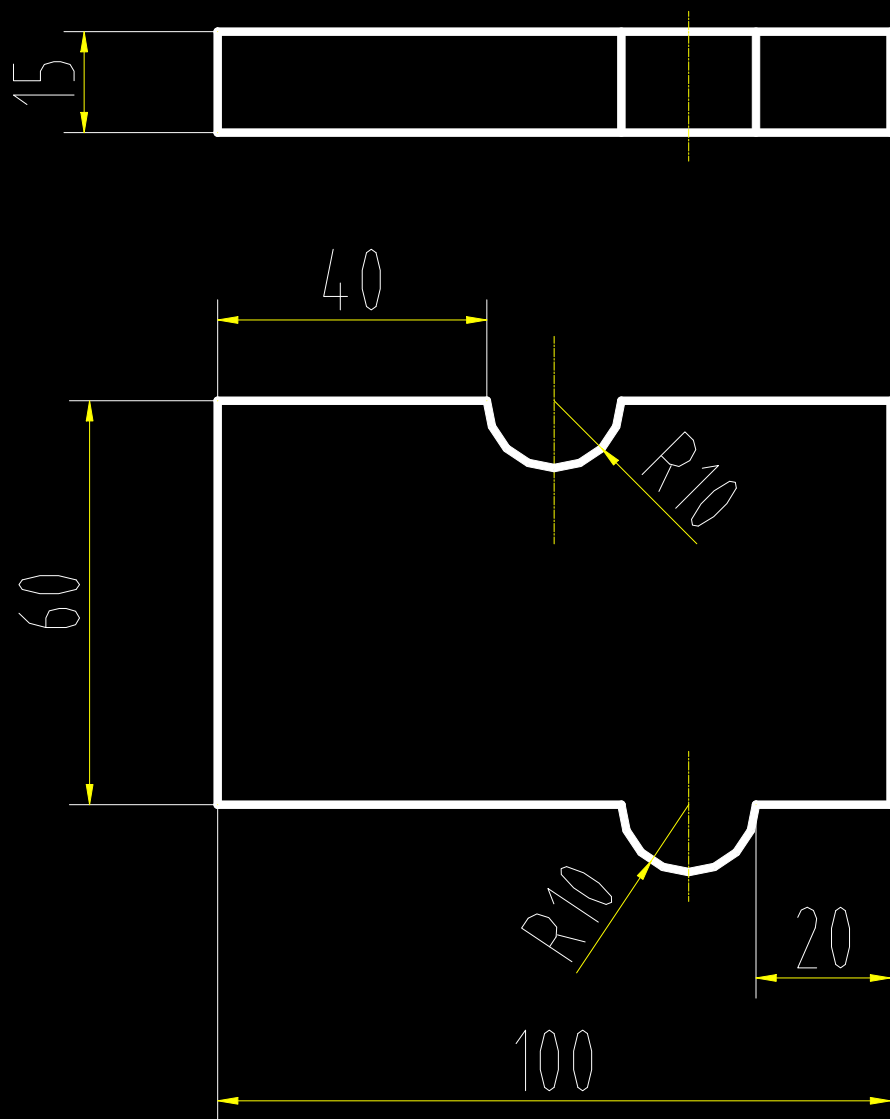


(2) 利用子程序调用实现分层加工。

如图所示零件

- ✓ 用 $\phi 8$ 立铣刀
- ✓ 粗铣外轮廓
- ✓ 深度方向分层铣削，分三次完成。（暂不考虑装夹）

7、子程序调用编程举例（三）



(3)利用子程序调用实现
分层、分次加工。

如图所示零件

- ✓ 用 $\phi 8$ 立铣刀
- ✓ 粗、精铣外轮廓
- ✓ 粗铣深度方向分三次完成。
- ✓ 精铣一次下刀完成
(暂不考虑装夹)

二、简化编程指令

1、镜像功能G24, G25

➤ 指令功能:

- 当工件(或某部分)具有相对于某一轴对称的形状时,可以利用镜像功能和子程序的方法,简化编程.
- 镜像指令能将数控加工刀具轨迹沿某坐标轴作镜像变换而形成对称零件的刀具轨迹。
- 对称轴可以是X轴、Y轴 或 X、Y轴。

➤ 指令格式:

G24 X__Y__Z__ 建立镜像
(M98 P_)

G25 X__Y__Z__ 取消镜像
或 G25

➤ 指令说明:

- **建立镜像**由指令坐标轴后的坐标值指定镜 像位置（对称轴、线、点）
- G24、G25为模态指令，可相互注销，G25为缺省值。
- 有刀补时，先镜像，然后进行刀具长度补偿、半径补偿。

例如：当采用绝对编程方式时

✓ **G24 X-9.0**

表示图形将以 $X=-9.0$ 的直线（//Y轴的线）作为对称轴，

✓ **G24 X6.0 Y4.0**

表示先以 $X=6.0$ 对称，然后再以 $Y=4.0$ 对称，两者综合结果即相当于以点（6.0，4.0）为对称中心的原点对称图形。

✓ **G25 X0**

表示取消前面的由**G24 X__** 产生的关于Y轴方向的对称

镜像指令编程

主程序

%0008

G92 X0 Y0 Z25.0

G90 G17 G00 Z5.0 M03

M98 P100 加工图1

G24 X0 坐标变换

M98 P100 加工图2

G24 Y0

M98 P100

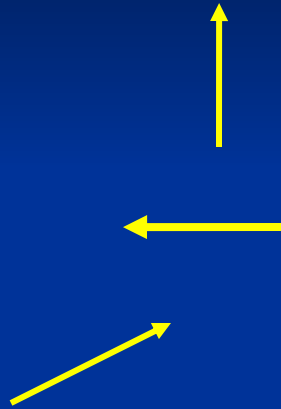
G25 X0

M98 P100

G25 Y0

Z25.0 M05M30

镜像指令编程



子程序

%100

G41 X10.0 Y4.0 D01

Y5.0

G01 Z-28.0 F200

Y30.0

X20.0

G03 X30.0 Y20.0 R10.0

G01 Y10.0

X5.0

G00 Z5.0

G40 X0 Y0

M99

2、旋转变换功能G68， G69

➤ 指令功能:

- 该指令可使编程图形按照指定旋转中心及旋转方向旋转一定角度。
- 通常和子程序一起使用，加工旋转到一定位置的重复程序段。

➤ **格式** G17 G68 X__Y__P__

G18 G68 X__Z__P__

坐标旋转功能

G19 G68 Y__Z__P__

G69

取消坐标旋转功能

其中：

X、Y、Z 是旋转中心的坐标值；

P为旋转角度，单位是 ($^{\circ}$)， $0 \leq P \leq 360.0^{\circ}$

逆时针旋转时为“+”， 顺时针旋转时为“-”

在有刀具补偿的情况下，先进行坐标旋转，然后才进行刀具半径补偿、刀具长度补偿。

在有缩放功能的情况下，先缩放后旋转。

主程序

%0009

G92 X0 Y0 Z25.0

G90 G17 G00 Z5.0 M03

M98 P100

G68 X0 Y0 P90.0

M98 P100

G69

G68 X0 Y0 P180.0

M98 P100

G69

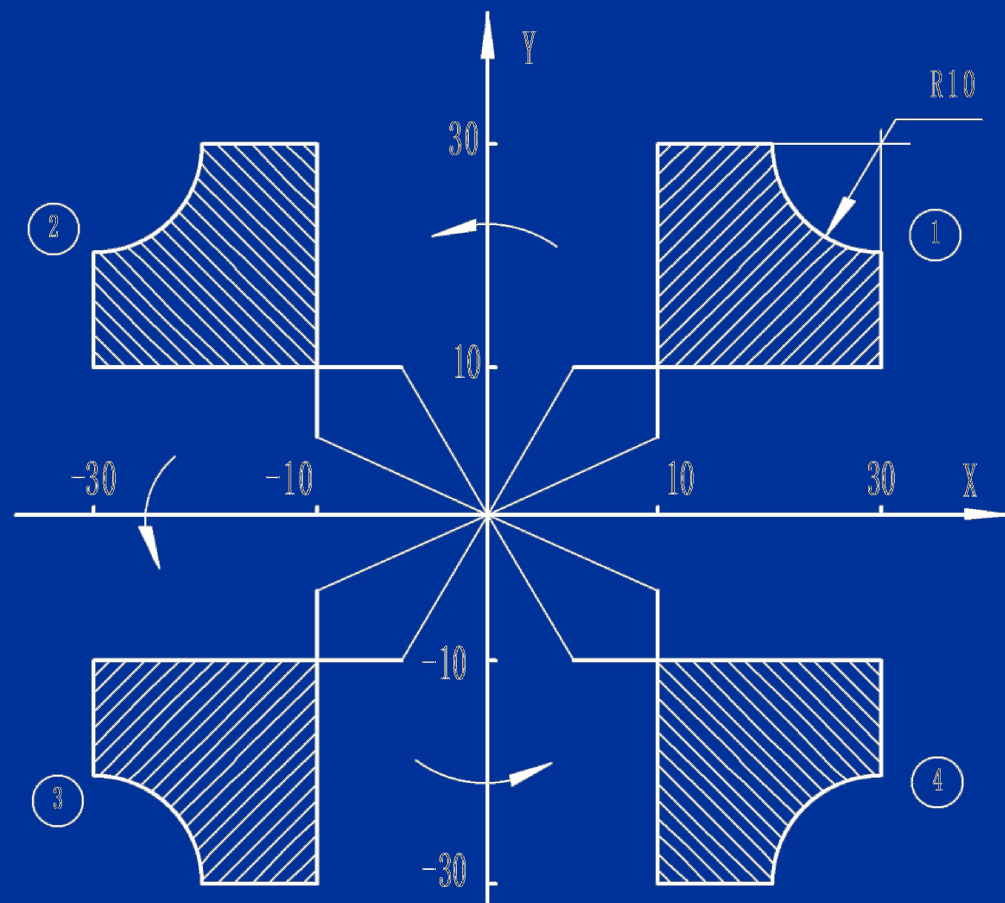
G68 X0 Y0 P270.0

M98 P100

G69

Z25.0 M05M30

旋转指令编程



将基本图形分别旋转
90°180°270°

3、缩放功能G50, G51

➤格式: G51 X_Y_Z_P_ 缩放开

(M98 P_)

G50 缩放关

- 其中: X、Y、Z给出缩放中心的坐标值, P后跟缩放倍数。
- G51既可指定平面缩放, 也可指定空间缩放。
- G51、G50为模态指令, 可相互注销, G50为缺省值。
- 有刀补时, 先缩放, 然后进行刀具长度补偿、半径补偿。

缩放指令编程

使用缩放指令可实现同一程序加工出形状相同，尺寸不同的工件。

主程序

```
%0007
```

```
G92 X0 Y0 Z25.0
```

```
G90 G00 Z5.0 M03
```

```
    G01 Z-18.0 F100
```

```
M98 P100
```

```
    G01 Z-28.0
```

```
G51 X15.0 Y15.0 P2
```

```
M98 P100
```

```
G50
```

```
G00 Z25.0 M05 M30
```

缩放指令编程

[返回上层](#)

使用缩放指令可实现同一程序加工出形状相同，尺寸不同的工件。

子程序

%100

G41 G00 X10.0 Y4.0 D01

G01 Y30.0

X20.0

G03 X30.0 Y20.0 R10.0

G01 Y10.0

X5.0

G40 G00 X0 Y0

M99

