

计算机绘图基础

司徒妙年

陈星铭

编著

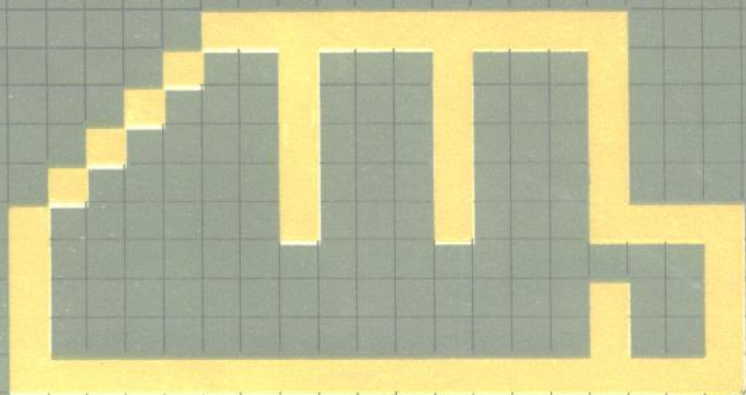
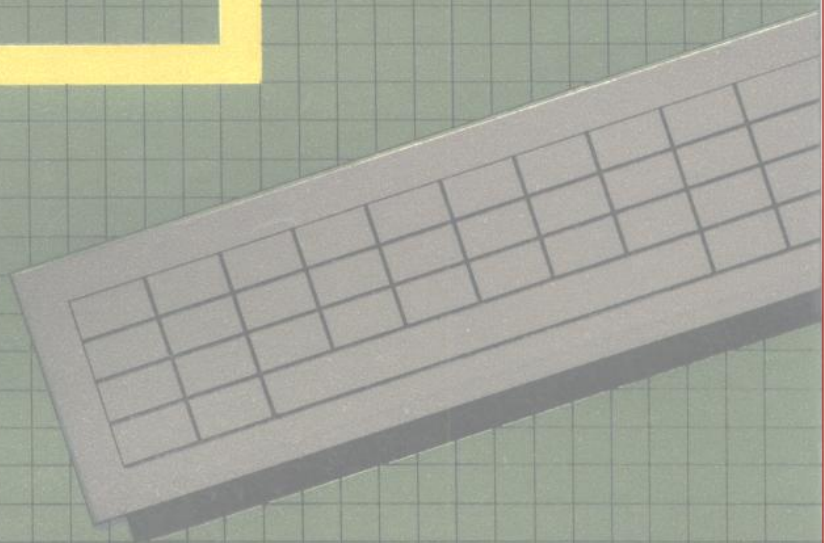
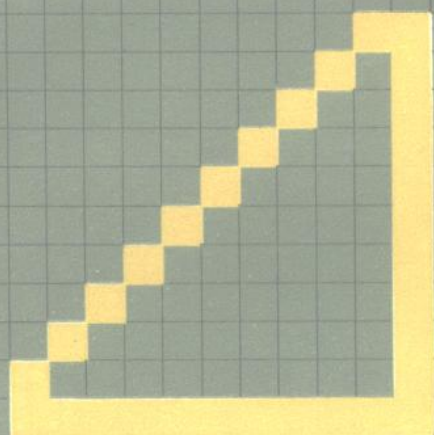
TH126

版社

计算机绘图基础

司徒妙年
陈星铭

编著



同济大学出版社

计算机绘图基础

司徒妙年 陈星铭 编著

同济大学出版社

内 容 提 要

本书根据 1994 年修订的《高等工业学校画法几何及工程制图课程教学基本要求》中有关“计算机绘图”的教学内容编写。书中介绍了有关图形生成原理、计算机绘图系统及绘图指令介绍;二维图形绘图程序设计、二维图形的变换方法、交互式计算机绘图软件介绍等内容。本书理论联系实际,有 GB 国标图纸幅面和工程图样等子程序设计、统计图形绘制、平面曲线和 Bézier 曲线设计等。各章后均附有练习,可供各校选用。

本书是一部介绍计算机绘图基础知识的教材,可供高等学校工科各专业使用,也可供职工业余大学、函授大学、电视大学等有关专业选用,并可供各类工程技术人员参考。

责任编辑 司徒妙龄
封面设计 李志云

计 算 机 绘 图 基 础
司徒妙年 陈星铭 编著
同济大学出版社出版
(上海四平路 1239 号)
新华书店上海发行所发行
同济大学印刷厂印刷

开本: 787×1092 1/16 印张: 9 字数: 230 千字
1995 年 10 月第 1 版 1995 年 10 月第 1 次印刷
印数: 1—3000 定价: 15.00 元
ISBN 7-5608-1466-2/TP·168

前 言

本书是根据国家教委委托“高等工业学校画法几何及工程制图课程教学指导委员会”1994年修订的《画法几何及工程制图课程教学基本要求》(以下简称《教学基本要求》)而编写的。编写过程中作了以下几方面的考虑:

第一,为了适应计算机绘图教学的发展,新修订的《教学基本要求》将计算机绘图部分扩展为20学时的计算机绘图基础。教学中迫切需要这部分内容的相应教材。故此,本书编写首先满足《教学基本要求》所提出的相应内容和要求,并考虑到机械类、土建类、非机械类各专业的需求,将这部分内容单独成书。

第二,内容安排上,力求按学生的认识规律,阐述计算机绘图程序编制的基本原理和方法。考虑到 FORTRAN 语言使用的普遍性和 C 语言的发展,本书大部分程序同时采用 FORTRAN 语言和 C 语言编写,以适应学生的学习需要。

第三,在满足《教学基本要求》的前提下,本书又适当加深、加宽内容。特别是对一种常用软件包(本书选用 Auto CAD)的操作及其应用作了较详细的介绍,可按不同专业的具体情况作相应的增删或选学。

各章后均有练习,可按需选做。

本书由上海交通大学卞博教授审定,他对本书的编写提出了许多宝贵意见,在此表示衷心的感谢。此外,本书在编写过程中还参考了一些有关书籍,特向有关的编著者表示衷心的感谢。参考书目列于书末。

本书第一章、第四章由司徒妙年编写;第二章、第三章由陈星铭编写。

由于计算机绘图课程教材建设的工作刚起步,更限于我们的水平,书中难免存在缺点和错误,恳请使用本书的师生和有关同志批评指正。

作者

1995.5.

目 录

第一章 计算机绘图系统及基本指令

§ 1-1 概述	(1)
§ 1-2 微机绘图系统的组成	(1)
§ 1-3 计算机图形显示及基本绘图指令	(2)
练习	(16)

第二章 二维简单图形绘图程序设计

§ 2-1 绘图程序的构成	(17)
§ 2-2 二维图形子程序的编制	(21)
§ 2-3 平面曲线子程序的编制	(27)
练习	(34)

第三章 二维图形变换的方法

§ 3-1 图形变换的基本概念	(36)
§ 3-2 二维图形的几何变换	(37)
§ 3-3 二维图形变换的程序设计	(43)
练习	(46)

第四章 交互式计算机绘图软件 Auto CAD 介绍

§ 4-1 Auto CAD 简介及启动	(48)
§ 4-2 绘图的准备工作	(53)
§ 4-3 用 Auto CAD 命令画简单的平面图形	(56)
§ 4-4 图形编辑命令	(69)
§ 4-5 图形的显示控制及辅助绘图工具	(84)
§ 4-6 图层和块	(89)
§ 4-7 尺寸标注与剖面线	(98)
§ 4-8 正等测图的绘制	(107)
§ 4-9 用 Auto CAD 绘图举例	(110)
练习	(117)

附录一 Auto CAD 命令摘录	(124)
附录二 点标记符号及其代码	(127)
附录三 Auto CAD 标准图案库摘录	(128)
附录四 Auto CAD 显示器配置方法	(130)
附录五 Auto CAD 的图形输出简介	(131)

第一章 计算机绘图系统及基本指令

§ 1-1 概 述

由于计算机科学和电子技术的飞速发展,作为工程图学与计算机科学相互结合的产物——计算机图形学,近年来得到了迅速发展和广泛应用,并且成为计算机领域中一支生力军。可以预见,随着计算机的日益发展与普及,计算机绘图必将取代用圆规和直尺的人工绘图,就像当年计算器取代计算尺一样。对于每一个工科学生来说,都必须掌握计算机绘图的基本原理和基本方法。

计算机绘图(Computer Graphics,简称CG)是计算机辅助设计(Computer Aided Design,简称CAD)和计算机辅助工程(Computer Aided Engineering)的重要组成部分。由于微型计算机技术的不断发展及其成本的不断下降,计算机绘图成为新技术革命的有力工具。各种高级的、独立于设备的图形软件包的出现,更使计算机绘图的成果得到迅速的推广和使用。

计算机绘图的基本原理,是将空间物体的几何特性归纳为数学模型,然后编成绘图程序输入计算机,通过计算机绘图系统,将其显示在屏幕上或绘制在图纸上。

计算机绘图可分为两种:

1. 用户使用高级语言中的绘图语句或绘图函数,编写成绘图源程序输入计算机,然后由计算机处理程序,输出图形。
2. 用户使用现成的绘图软件包。使用时先把绘图软件包装入计算机,然后用软件包内设计好的一系列绘图命令输入给计算机,计算机处理命令后输出图形。

现成的绘图软件包简单易学,易于推广,便于使用。但并不是说有了现成的绘图软件包,就不必再自己设计绘图程序了。首先,绘图软件包本身就是由一系列的绘图源程序构成。其次,当我们要绘制一些变化有规律的图形或改变某些参数就可从一个图形变换到另一个图形时,显然是使用自己设计的源程序来得简捷方便。此外,为了进行绘图软件的二次开发,也需要掌握图形编程方法及处理技术。

§ 1-2 微机绘图系统的组成

一、主机

主机主要由中央处理器(CPU)和内存贮器构成。微机的功能在很大程度上取决于中央处理器,因此习惯上用CPU档次来概略表示微机的规格,现在用得较多的机种有286,386和486。

二、外存贮器

目前最常用的外存贮器是磁盘,包括硬盘和软盘。一般微机使用的硬盘的容量为10M到

几百兆,软盘有 5 吋盘和 3 吋盘两种,5 吋盘的容量又有 360K(低密盘)和 1.2M(高密盘)两种,3 吋盘的容量为 1.44M。不同的磁盘分别由机内相应的磁盘驱动器进行读写。

三、输入设备

微机的主要输入设备是标准的 ASCII 码键盘,此外常用的还有鼠标器、数字化仪、图形输入板、图形扫描仪等。

四、显示器

显示器用来显示、输出各种图形、数据和信息,是微机必不可少的外部设备之一。显示器分黑白和彩色两种,为了方便起见,也可设置两台显示器,一台用来显示图形,另一台用来输出各种数据、指令及程序。

五、其他输出设备

最常用的有打印机和绘图机。打印机可用来打印程序清单、数据或屏幕上的图像。绘图机是绘图系统的主要输出设备,它工作时接受主机传送来的绘图信息,在图纸上绘出图形。

§ 1-3 计算机图形显示及基本绘图指令

本节将简要介绍用 Turbo C 语言和 FORTRAN 语言在 IBM-PC 微机上的图形显示方法及基本绘图指令。

一、Turbo C 语言的图形显示及基本绘图指令

Turbo C 2.0 提供了一个具有丰富图形函数的图形库,这些函数包含在库文件 GRAPHICS.LIB 中,编写程序时,应先嵌入该函数库对应的头文件 graphics.h。这里介绍的是 Turbo C 2.0 的部分图形显示及绘图函数。运行 Turbo C 需 DOS 2.0 版本或更高版本的支持。

1. initgraph() 初始化图形系统

功能:装入图形驱动程序,并将系统设置为图形模式。

调用格式: `initgraph(int far *gdriver, int far *gmode, char far *path);`

参数说明: *gdriver ——指定要调用的图形驱动程序;

*gmode ——说明使用何种初始图形模式;

*path ——指定进入图形驱动程序的路径。

例 1-1

```
#include <graphics.h>
main()
{
    int gdriver, gmode;
    gdriver = VGA; /* 设置显示器为 VGA */
    gmode = VGAHI; /* 选用 VGAHI 图形模式,分辨率为
                    640×480,16 色显示 */
    initgraph(&gdriver, &gmode, "c:\\TC"); /* 初始化图形系统 进入驱动程序的路
```

： 径为 C 盘中的 TC 目录 * /

若不知所用图形显示器的种类,则可把上例中的两句赋值语句

gdriver=VGA;

gmode=VGAHI;

改为 gdriver=DETECT;

把 initgraph()函数写成 initgraph(&gdriver,&gmode,path); 它就会自动地搜索显示器类型和显示模式。

2. closegraph() 退出图形系统

功能:关闭图形系统,恢复初始的视频模式,释放图形驱动程序和图形方式所占用的内存。

调用格式: closegraph(void);

3. setbkcolor() 设置背景色

功能:将背景设置成参数 color 所指定的颜色。

调用格式: setbkcolor(int color);

参数说明: color 为 0~15 的整数,一个数值对应一种背景色,见表 1-1。

表 1-1

color 数值	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
颜色	黑	蓝	绿	青	红	洋红	棕	淡灰	深灰	淡蓝	淡绿	淡青	淡红	淡紫	黄	白

4. setcolor() 设置画线颜色

功能:将当前画线颜色设置成参数 color 所指定的颜色。

调用格式: setcolor(int color);

参数说明: color 的取值范围与数值所对应的颜色,与 setbkcolor()中相同。

例 1-2

```
#include<graphics.h>
main( )
{
    int driver,mode;
    driver=DETECT;          /* 规定自动搜索显示器类型和显示模式 */
    initgraph(&driver,&mode,"c:\\\\TC");
    setbkcolor(15);         /* 设置背景色色号为 15,白色 */
    setcolor(4);            /* 设置画线色号为 4,红色 */
    circle(200,150,50);     /* 画直径为 50、圆心在 x=200,y=150 处的圆,红色 */
                             /* */
    setcolor(14);           /* 设置画线色为 14,黄色 */
    circle(200,150,100);    /* 画圆心 x=200,y=150,直径为 100 的圆,黄色 */
                             /* */
    getch();               /* 等待键盘输入任一字符 */
    closegraph();          /* 退出图形系统 */
}
```


5. line() 画直线

功能: 在指定的两点(x0,y0)和(x1,y1)之间画一条直线,当前点(CP)不变。

调用格式:line(int x0,int y0,int x1,int y1);

参数说明:x0,y0,x1,y1 为屏幕的 x 向和 y 向坐标值。屏幕坐标的原点(0,0)在屏幕的左上角,x 向右为正,y 向下为正。初始状态下,CP 为(0,0)。

6. lineto() 画直线

功能:从当前位置(CP)到指定点(x,y)画一条直线,然后把(x,y)作为当前点(CP)。

调用格式:lineto(int x,int y);

参数说明:x,y 的含义及取值方法同 line()。

7. linerel() 画直线

功能:从当前点(CP)到相对 CP 坐标为(dx,dy)的点画一条直线,同时 CP 移到该点。

调用格式:linerel(int dx,int dy);

参数说明:dx,dy 取值方法与 line() 同,由于 dx,dy 是相对坐标,所以 dx,dy 取正值时,沿 x,y 的正向画直线,dx,dy 取负值时则沿 x,y 的负向画直线。

8. moveto() 移笔

功能:将画笔从当前点(CP)移到指定位置(x,y)处。

调用格式:moveto(int x,int y);

参数说明:x,y 为绝对坐标值,取值方法同 line()。

9. moverel() 移笔

功能:将画笔从当前点(CP)移到相对坐标为 dx,dy 之点。

调用格式: moverel(int dx, int dy);

参数说明: dx,dy 分别为 x 向和 y 向上的位移量,为正值时位移沿 x,y 轴的正向移动,为负值时位移沿负向移动。

例 1-3

```
#include <graphics.h>
#include <stdio.h>
main( )
{
    int driver, moed;
    driver=VGA; mode=VGAHI;
    initgraph(&driver, &mode, " ");
    line(50,300,250,300); /* 从(50,300)向(250,300)画直线 */
    printf("Current point station is:%d,%d\n", getx(), gety()); /* 显示 CP 坐标值 */
    moveto(250,300); /* 把 CP 移到(250,300)处 */
    lineto(250,100); /* 从(250,300)向(250,100)画直线 */
    linerel(-200,0); /* 画直线,x 向左位移量 200,y 向为 0 */
    lineto(50,300); /* 从当前点 CP 向(50,300)画直线 */
    printf("Current point station is:%d,%d\n", getx(), gety()); /* 显示 CP 坐标值 */
    moverel (100,-100); /* 把 CP 沿 x 正向移动 100,沿 y 负向移 100 */
    printf("Current point station is:%d,%d\n", getx(), gety()); /* 显示 CP 坐标值 */
}
```

```

getch();
closegraph();
}

```

以上程序画了一个边长为 200 的正方形,最后把当前点(CP)移到正方形的中心。其间,三次调用 `getx()` 和 `gety()` 函数,在屏幕上显示不同的当前点的坐标值。`get()`, `gety()` 都是图形函数,其功能是返回当前点的 `x` 和 `y` 坐标值。

10. `arc()` 画圆弧

功能:以 (x,y) 为圆心, `radius` 为半径,从起始角 `stangle` 到终止角 `endangle` 画圆弧。

调用格式:`arc(int x,int y,int stangle, int endangle, int radius);`

参数说明: (x,y) 为圆弧所在圆之圆心,角度以度为单位,逆时针时为角度的正向,顺时针为负向,如图 1-1 所示。

11. `circle()` 画圆

功能:以 (x,y) 为圆心, `radius` 为半径画圆。

调用格式:`circle(int x,int y, int radius);`

参数说明: (x,y) 为圆心坐标值, `radius` 为圆之半径。

12. `ellipse()` 画椭圆

功能:以 (x,y) 为中心, `stangle` 和 `endangle` 为起、止角, `xradius` 和 `yradius` 为长、短半径画椭圆弧。

调用格式:`ellipse(int x, int y, int stangle, int endangle, int xradius, int yradius);`

参数说明:`x,y,stangle, endangle` 与画圆弧函数的参数意义、用法完全相同, `xradius` 为椭圆水平方向的半径, `yradius` 为垂直方向的半径。

例 1-4

```

#include <graphics.h>
main()
{
    int i,gdriver=VGA, gmode=VGAHI;
    initgraph(&gdriver,&gmode,"");
    setbkcolor(0);           /* 设置背景色为黑色 */
    setcolor(2);             /* 设置画笔色为绿色 */
    circle(320,240,100);     /* 画圆,圆心(320,240),半径 100 */
    setcolor(4);             /* 设置画笔色为红色 */
    ellipse(320,240,0,360,100,200); /* 画椭圆,横半径 100,竖半径 200,中心为
                                (320,240) */
    ellipse(320,240,0,360,200,100); /* 画椭圆,中心不变,横、竖半径分别为
                                200,100 */
    setcolor(14);            /* 设画笔色为黄色 */
    for(i=0;i<=3;i++)
        arc(320,240,90*i+15,90*i+60,80); /* 画圆弧,起、止角分别为 90*i+15 和 90
                                *i+60 */
}

```

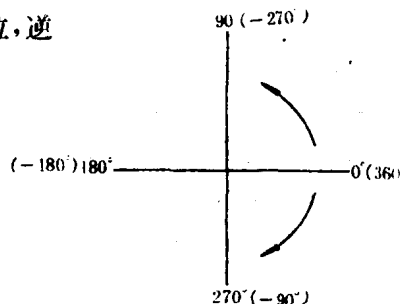


图 1-1

```

getch();
closegraph();
}

```

这段程序画了一个半径为 100 的圆,两个长短轴半径同为 100 和 200 的椭圆,一个长轴横放,一个长轴竖放,另又画了四段圆心角皆为 45°的圆弧。以上圆或圆弧的圆心皆为同一点。

13. rectangle() 画矩形

功能:画一给定对角点的矩形。

调用格式:rectangle(int left, int top, int right, int bottom);

参数说明:(left, top)为矩形的左上角点坐标,(right, bottom)为右下角点坐标。

14. drawpoly() 画多边形

功能:画封闭的多边形或折线。

调用格式:drawpoly(int numpoints, int far * polypoints);

参数说明:numpoints 为折线或多边形的顶点数,* polypoints 指向一整型数序列,由 2 * numpoints 个整数组成,每一对整数给出一个多边形顶点的 x,y 坐标,如果要画一封闭的 n 边的多边形,则必须将 n+1 个坐标点传给 * polypoints,其中第 n 个坐标点与第 0 个坐标点相同。

例 1-5

```

#include <graphics.h>
main()
{
    int gdriver, gmode;
    int rhombus[]={100,100,170,50,240,100,170,150,100,100}/* 整型数序列定义 */
    gdriver=DETECT;
    initgraph(&gdriver,&gmode," ");
    rectangle(50,50,800,600);          /* 画矩形 */
    drawpoly(5,rhombus);               /* 画多边形,此多边形为
                                         一菱形 */

    getch();
    closegraph();
}

```

15. setlinestyle() 设置线宽和线型

功能:设置当前图线的线型、模式和宽度。

调用格式:setlinestyle(int linestyle, unsigned upattern, int thickness);

参数说明:linestyle 说明了画后续线的线型,图形库中已定义的线型如表 1-2 所示。调用时,可直接引用其名称,也可以输入其相应的值,二者效果是一致的。

upattern 用 16 进制的数值给出用户定义的线型,其时 linestyle 的值必须为 4,即相当于用户选择 USERBIT-LINE 这一线型。该 16 位值中的每一位等于一个像素,哪一位为 1,则该位的像素用当前绘图颜色绘出。当 linestyle 的值不为 4, upattern 参数仍是需要的,只是不起作用,为方便起见,一般给其赋值 0。

thickness 指明以后所画线的宽度,线的宽度有两种,如表 1-3 所示。调用时可直接引用其值,也可引用其名称。

表 1-2

名 称	值	说明及图例
SOLID-LINE	0	实线 
DOTTED-LINE	1	点线 
CENTER-LINE	2	中心线 
DASHED-LINE	3	虚线 
USERBIT-LINE	4	用户定义的线型

表 1-3

名 称	值	说 明
NORM-WIDTH	1	1 个像素宽
THICK-WIDTH	3	3 个像素宽

例 1-6

```

#include <graphics.h>
#include <stdio.h>
main()
{
    int gd, gm, s, x0, y0, i;
    gd = VGA; gm = VGAHI;
    initgraph(&gd, &gm, "c:\\TC");
    printf("Input x0, y0, s, please! \n");
    scanf("%d, %d, %d", &x0, &y0, &s);
    setbkcolor(0);
    setlinestyle(0, 0, 1);
    for (i = 1; i <= 15; i++)
    {
        setcolor(i);
        square(x0, y0, s);
        x0 = x0 - 10;
        y0 = y0 - 10;
        s = s + 20;
    }
    getch();
    closegraph();
}

square(x1, y1, s)
int x1, y1, s;
{
    int x, y;
    x = x1 + s;
    y = y1 + s;
    moveto(x1, y1);
    lineto(x, y1);
    lineto(x, y);
    lineto(x1, y);
}

```

```

        lineto(x1,y1);
    }

```

以上程序画了 15 个颜色依次变化、边长依次增加 20 的正方形。printf() 函数调用句要求用户输入画最小的正方形的起点和边长。setlinestyle(0,0,1); 句给定了画正方形的线型(实线)和线宽(细)。循环体中的 setcolor(i); 使画出的正方形的颜色随 i 的增加而变化。X0=X0-10; 和 Y0=Y0-10; 两句, 使得画正方形的起点每次向左、向上移动 10, s=s+20; 句使正方形边长每次增加 20。square(x1,y1,s); 是函数调用, 该自编函数放在上程序之后。

二、程序的装入、编辑和运行(TC 集成环境简介)

Turbo C 提供了一个完整的交互式集成开发环境, 这是一个高效、快速的编译程序。用户可通过一个简单清晰的主屏幕, 使用编辑、调试、编译、连接等功能, 装入和运行文件。下面简要介绍如何使用主屏幕装入和编辑、运行源程序。

1. 进入 TC

如果已在硬盘 C 安装了 Turbo C, 以后每次要进入 TC, 可按如下步骤操作:

C>CD\TC\ (带下划线部分为用户输入, \ 表示回车键)

C>TC\

进入 TC 环境后, 屏幕出现产品版本信息框, 按任意键, 即显示完整的 TC 主屏, 如图 1-2 所示。

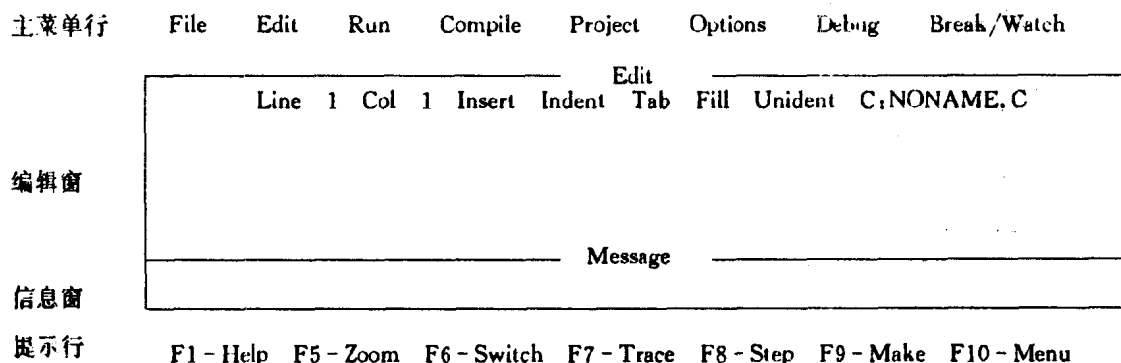


图 1-2 集成开发环境 TC 主屏

2. 装入源程序文件

把光标移至主菜单行的 File 处, 回车, File 项下将出现一下拉式菜单, 可用 ↑、↓ 键移动光标选择其中的一项。如要装入新的文件, 可选 Load 或 New。选 Load 项, 可直接键入文件名后回车, 屏幕的右上方会显示该文件名, 并自动加上扩展名 C。选择 New 项, 屏幕右上方自动显示缺省文件名 NONAME.C, 可在存盘时改变文件名。写入文件名后, 就可在编辑窗输入源程序文件了。如要调出已存盘的文件, 只要在选择 Load 项后出现的提示框里直接键入文件名后回车; 如忘了存盘的文件名, 可不键入文件名而直接回车, TC 会自动列出当前目录下所有具扩展名 .C 的文件, 此时可移动光标选取所需文件。

3. 建立可执行程序(编译、连接)

用光标选择主菜单行的 Compile, 即可把指定的文件或最后装入编辑窗的文件进行编译和连接。如程序有错误, 屏幕下方的信息窗会出现有关的错误信息和警告信息, 同时光标出现在程序中的出错处提醒用户修改。如程序编译和连接成功, 编译窗口会给出信息 "Press Any

Key”。完成编译和连接最简单的方法就是按 F9 键。

4. 运行程序

建立了可执行程序之后,欲运行程序可选择主菜单中的 Run。如选择 Run 的下拉式菜单中的 Run,由于运行速度极快,用户屏幕几乎只是一闪就又回到了 TC 主屏,因此如要观察程序的输出内容,可选择 Run 下拉式菜单中的 User Screen,该项可把主屏幕切换到用户屏幕。如欲返回主屏幕,任按一键即可。

5. 常用的 TC 功能键

所谓“功能键”就是只要按下它就能完成某一功能的键,显然,使用功能键比使用菜单快捷得多。表 1-4 列出了 TC 中常用的功能键。表中,Ctrl - F1 即表示先按下 Ctrl 键不放,再按 F1 键,余可类推。File/Save 表示主菜单中 File 的下拉式菜单中的 Save 项,余同。

TC 常用功能键

表 1-4

功能键	功 能	相当的菜单命令
F1	显示当前位置的帮助信息	
F2	保存编辑程序里的当前文件	File/Save
F3	装入一个文件	File/Load
F4	程序运行到光标所在行	Run/Go to cursor
F5	放大/缩小窗口,即在全屏幕和分割屏幕模式之间切换活动窗口	
F6	切换编辑窗与观察窗或信息窗	
F7	调试模式下运行程序,跟踪进函数内部	Run/Trace into
F8	调试模式下运行程序,跳过函数调用	Run/Step over
F9	执行 Make,生成 EXE 文件(可执行文件)	Compile/Make EXE file
F10	切换菜单和编辑窗口	
Ctrl - F9	运行程序或编译、连接并运行程序	Run/Run
Ctrl - F2	中止当前的调试,释放用户程序所占内存,并关闭所有打开的文件	Run/Program Reset
Alt - F3	选取文件并装入	File/Pick
Alt - F5	切换 TC 主屏幕与用户屏幕	Run/User screen
Alt - F6	切换活动窗口的内容	
Alt - B	进入 Break/watch 菜单,相当于用光标选取 Break/watch 主菜单	
Alt - C	进入 Compile 菜单	
Alt - E	进入 Edit 菜单	
Alt - F	进入 File 菜单	
Alt - R	进入 Run 菜单	
Alt - X	退出 TC,返回 DOS	File/Quit

6. 存盘

可选择 File/Save 把当前编辑的文件存盘,F2 键具有同样的功能。也可选择 File/Write to 把当前编辑的文件换名存盘。

7. 编辑器的使用

TC 编辑器的命令与 Wordstar 兼容。最常用的编辑命令如下:

←,→	左、右移一字符
↑,↓	上、下移一行
PgUp, PgDn	上、下翻一页
Ctrl - T	删除光标所在单词

Ctrl - Y	删除光标所在行
Ctrl - KB	标记块头
Ctrl - KK	标记块尾
Ctrl - KC	在光标位置处复制块
Ctrl - KV	将块移至光标位置处
Ctrl - KY	删除块
Ctrl - KH	隐藏/显示块标志

三、FORTRAN 语言的图形显示及基本绘图指令

FORTRAN 语言本身并不具备绘图能力,一般都是用 CALL 语句调用为 FORTRAN 语言开发的绘图功能子程序。这里介绍的是华东工业大学的 SIMECCS 绘图系统中的部分图形显示及基本绘图指令。该系统要求 DOS2.0 以上版本的操作系统的支持,配有一张系统软盘。该系统中的子程序分属如下几个模块:

- SRA ——基本图素显示模块;
- SRB ——基本图形绘图模块;
- SRC ——尺寸标注、零件结构要素、基本子图形、剖面线绘制模块;
- SRD ——几何元素交切计算模块;
- SRE ——图形变换、三视图与轴测图绘制、凸体消隐处理、二维图形剪取模块;
- SRF ——曲线拟合与曲线插值模块。

本书将介绍 SRB 模块中的部分子程序的调用方法。

1. INIT 命令

功能:启动绘图系统,选择图形输出状态。

调用格式: CALL INIT

参数说明:此命令无参数。执行此命令后,屏幕上将出现 三行提示,要求输入 1,0 或-1。若输入 0,屏幕上显示图形;输入 1,显示并存储图形;输入-1,可接通绘图仪绘制图形。选择了图形输出状态后,屏幕上又出现提示,要求输入比例系数 S 和平移量 X0,Y0,S>1,放大,S<1,缩小。

2. SCA 命令

功能:设置比例及平移参数。

调用格式: CALL SCA(S,X,Y)

参数说明: S ——比例系数。S>1,放大,S<1,缩小;

X,Y ——平移参数。

3. FINISH 命令

功能:关闭绘图系统。

调用格式: CALL FINISH

参数说明:无形参。

4. MOVE 命令

功能:生成直线段或定位。

调用格式: CALL MOVE(X,Y,ND)

参数说明: X,Y ——指定点的坐标;

ND ——笔状态码, 整型数;
 ND=2, 从当前点至指定点画直线;
 ND=3, 从当前点抬笔至指定点。

5. CIR1 命令

功能: 画圆或圆弧。

调用格式: CALL CIR1(XC, YC, R1, AL, GA, ND)

参数说明: XC, YC ——圆心坐标;

R1 ——半径, $R1 > 0$, 逆时针画圆弧, $R1 < 0$, 顺时针画圆弧;

AL ——起始角(度);

GA ——终止角(度);

ND ——笔状态码;

ND=2, 从当前点至圆弧起点画直线;

ND=3, 从当前点抬笔至圆弧起点。

见图 1-3。

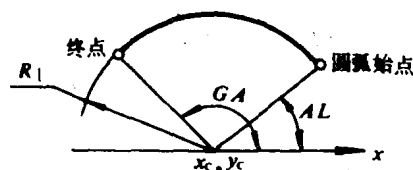


图 1-3 CIR1 参数

6. CIR2 命令

功能: 画加宽线圆弧。

调用格式: CALL CIR2(XC, YC, R1, K, AL, GA)

参数说明: K ——线宽系数, $K=1\sim3$ 。

7. LINEA 命令

功能: 画加宽直线。

调用格式: CALL LINEA(XT, YT, XS, YS, K)

参数说明: XT, YT ——直线起点坐标;

XS, YS ——直线终点坐标;

K ——线宽系数, $K=1\sim3$ 。

8. LINEB 命令

功能: 画虚线、点划线、双点划线。

调用格式: CALL LINEB(XT, YT, XE, YE, D1, D2, D3, K, IP)

参数说明: XT, YT ——起点坐标;

XE, YE ——终点坐标;

D1, D2, D3 ——虚线、点划线、双点划线的长度与间隔(见图 1-4)。

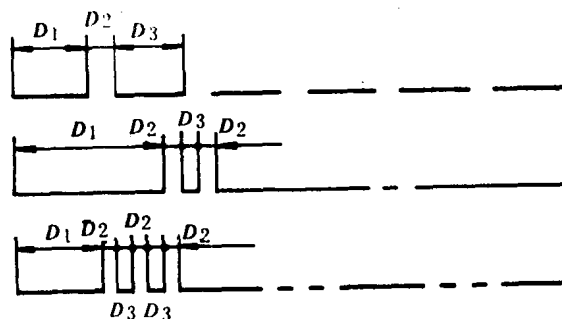


图 1-4 D1, D2, D3 参数

IP ——状态码;

IP=1,画虚线;
 IP=2,画点划线;
 IP=3,画双点划线;
 K——线宽系数,K=1,2。

9. CIR3 命令

功能:画点划线圆弧。

调用格式:CALL CIR3(XC,YC,R,D1,D2,D3,AL,GA)

参数说明:D1,D2,D3——点划线圆弧段长度与圆弧间隔,与 LINEB 中的 D1,D2,D3 类同。

其他参数同 CIR1。

10. CIR4 命令

功能:画虚线圆弧。

调用格式:CALL CIR4(XC,YC,R,K,D1,D2,AL,GA)

参数说明:K——线宽系数,K=1,2。

其他参数同 CIR3。

11. TEXTA 命令

功能:写字符串。

调用格式:CALL TEXTA(X,Y,H,AL,CH,NH,K)

参数说明:X,Y——字符串第一个字符左下角坐标;

H——字高;

AL——字符串与 X 轴正向夹角;

CH——‘字符串’;

NH——字符串个数,整型数;

K——状态码;

K=1,正体绘制

K=-1,斜体绘制

} 屏幕上只能水平显示,绘图仪上可以斜体输出。

12. NEWPEN 命令

功能:换笔。

调用格式:CALL NEWPEN(N)

参数说明:N——笔号,N=1~6(对应六枝笔)。

例 1-7 用 FORTRAN 语言编写绘制图 1-5 所示图形的源程序。

§ STORAGE: 2

```

program example 7
dimension xc(10),yc(10),r(10),al(10),ga(10)
data xc/2 * 220. ,2 * 189.689,2 * 250.311,2 * 220. ,186. ,254. /,
* yc/175. ,105. ,122.5,157.5,122.5,157.5,2 * 140. ,2 * 25. /,
* r/6 * 8. ,21. ,60. ,2 * -10. /,al/8 * 0. ,180. ,90. /,ga/8 * 360. ,
* 90. ,0. /
call init
call newpen(1)
  
```