目　　录

§1 计算机绘图概述

一、为什么要学习计算机绘图

长期以来人们都是用手工进行绘图，这是一项繁琐、费时的劳动。计算机问世以后，这种局面有了改观，计算机能使工程设计人员从繁琐重复的计算和艰辛的设计绘图工作中解放出来。如果把人的聪明才智和创造能力与计算机的不怕繁琐和高速运算功能相结合，则能使设计绘图工作的效率提高十倍乃至百倍。因此，工程设计与制图的计算机化是设计制图发展的必然趋势。

目前，人们正在将计算机绘图系统与整个设计过程联系起来。在设计过程中，计算机及时地输出计算结果并显示图形，供设计人员审视修改，直到满意为止。这种设计称为计算机辅助设计（简称CAD）。它与计算机辅助制造（简称CAM）相结合构成设计、制图、制造的一体化系统（CAD/CAM系统）。计算机辅助设计进入最终阶段，得到满意的结果时，将设计的图形用屏幕或绘图机输出，这就是常称的计算机绘图（简称CG）。计算机绘图的硬件和软件是应用CAD/CAM技术必不可少的内容。计算机绘图的硬件不仅包括计算机本身，而且包括自动绘图设备。计算机绘图的软件包括研究图形显示和图形输出的程序等等。国外把计算机辅助设计、计算机绘图和计算机辅助制造视为工业生产和设计的革命性变革，并把该一体化

系统的内容列入工科大学的课程中，有很多大学开设了计算机绘图课。计算机技术在我国发展得也很快，计算机辅助设计、计算机绘图已相当广泛地用于许多行业，如飞机、船舶、汽车、机械工程、电气工程、建筑设计等。国内许多单位研制的软件质量很高，可与国外同类软件相媲美。这说明我国在这一新兴学科领域中是很有发展潜力的。普及推广计算机绘图技术，将会促进CAD/CAM技术的发展，并加速我国现代化的进程。

为适应设计工作与生产发展的需要，以便在将来的工作中取得成绩，工科院校学生学习计算机绘图是非常必要的。

学习计算机绘图的基本知识并不困难。读者在学习了BASIC 算法语言的基础上，学习本书，再经过上机实验，就能对计算机绘图有一定的了解。

下面以苹果Ⅱ微型计算机和SPL-400 型绘图机为例，简要介绍计算机绘图的初步知识。

二、苹果Ⅱ微型计算机绘图系统简介

苹果Ⅱ微型计算机绘图系统的硬件由三个部分组成：主机、键盘和显示器。它具有数学运算、数据处理和图形显示等功能。若再配上一些外部设备，如磁盘机、自动绘图机等，就能实现绘图软件的存贮和较好的图形输出。

苹果Ⅱ微型计算机绘图系统硬件如图 1 所示。

各部件的功能如下：

1. 主机　进行运算和数据处理，控制各外部设备。

2. 键盘　用以输入字符、数据、指令、程序，是人－机

对话的输入工具。

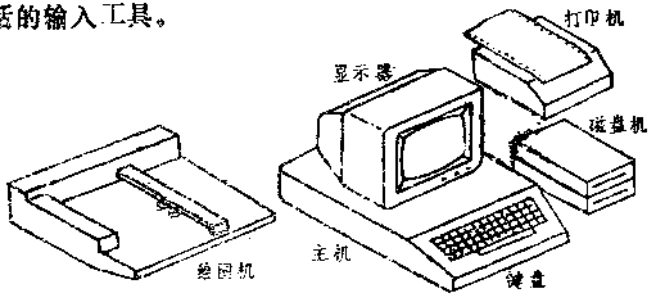


图 1

3. 显示器 用于显示字符、图形和程序等。

4. 打印机 能按要求打印出屏幕上显示的字符、图形和程序。

5. 绘图机 用于绘制较精确的图形。常见的绘图机有平板式和滚筒式两类。平板式绘图机工作时图纸平铺并固定在绘图机板面上，由可移动的画笔按绘图指令作X、Y方向移动绘出图形，如图2(a)所示。滚筒式绘图机工作时，将专用的绘

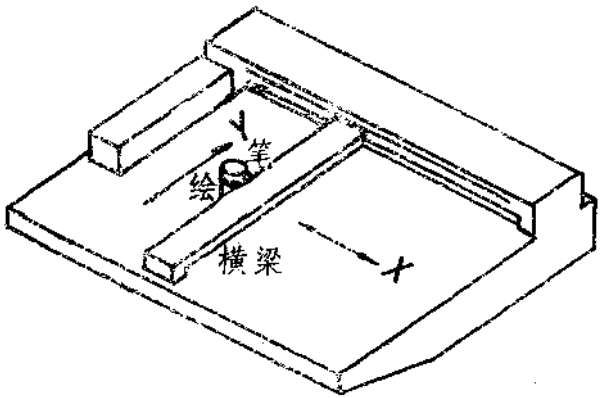


图2 (a)

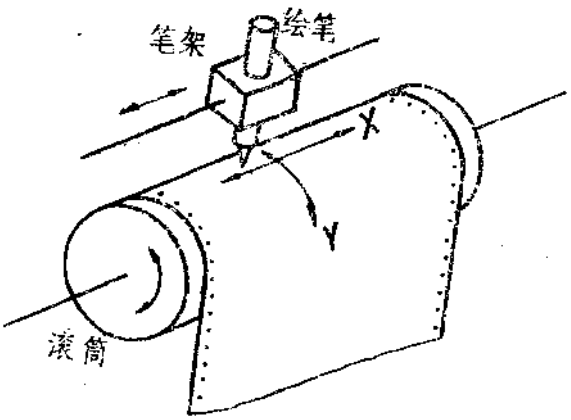


图2 (b)

图纸卷在一个能往复运动的滚筒上，绘图纸随滚筒转动实现Y方向移动，绘图机上的画笔只作X方向移动，从而实现绘图，如图2(b)所示。

6. 磁盘和磁盘机 是外存贮器，能把程序、数据等文件可靠地存贮起来，以备日后随时调用。

§2 图形显示和图形打印

一、高分辨率图形显示

（一）苹果Ⅱ微型计算机屏幕显示方式可分为文本显示与图形显示两种

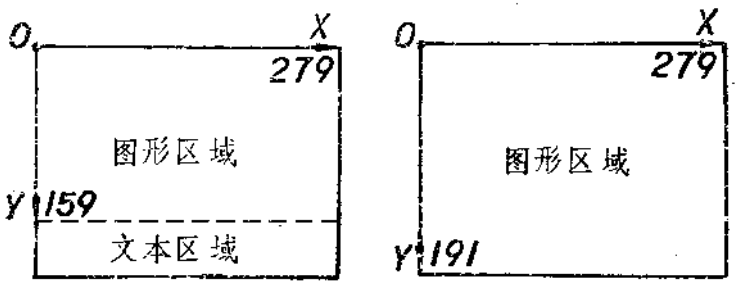
文本显示：用于显示文本、数据、符号等。

图形显示：用于显示图形。图形显示有低分辨率和高分辨率两种显示方式。

1．低分辨率图形显示　屏幕画面由40×40或40×48个象素（光点）组成。由于光点呈小方块状，不宜构成较精细的图形，故在工程制图中不使用。这里不作介绍。

2．高分辨率图形显示　有两种形式。

（1）图形与文本混合显示　这种形式能同时在屏幕上显示图形和文字。图形区域在屏幕上部，水平方向自左至右共280个光点，即X方向由0至279点；垂直方向自上而下共160个光点，即Y方向由0至159点，文本区域（显示文字、符号）在屏幕下部，可显示四行文字，如图3（a）所示。



（a）HGR　　（b）HGR2

图　3

图形文本混合显示时：
$0 \leq X \leq 279$；
$0 \leq Y \leq 159$。

全屏幕图形显示时：
$0 \leq X \leq 279$；
$0 \leq Y \leq 191$。

（2）全屏幕图形显示　这种形式只能在屏幕上显示图形，不显示文字，全部屏幕为图形区域。水平方向自左至右仍

为280个光点，即X方向由0至279点；垂直方向自上至下共192个光点，即Y方向由0至191点，如图3（b）所示。

3．屏幕坐标系　屏幕坐标系的原点定在屏幕的左上角，即X、Y都是零。X坐标由原点向右增大为正向，**Y坐标由原点向下增大为正向**，如图4所示。这一点与数学上常用的坐标系不同，应特别注意，否则会出错或将图画颠倒。

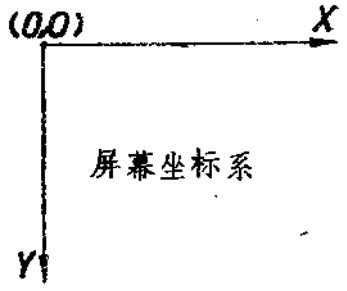


图4　屏幕坐标系

（二）几条常用的高分辨率图形显示指令

1．HGR　图形、文本混合显示形式指令。

2．HGR2　全屏幕图形显示形式指令。

3．HCOLOR　屏幕图形选用颜色指令。

对于单色显示器显示图形，取HCOLOR＝3时，表示选用图形颜色为白色。若取HCOLOR＝0表示选用黑色，即与屏幕底色相同，常用于“擦去”画过的图形。

4．HPLOT　显示点或直线的指令。

格式：

（1）HPLOT　X，Y（显示一点）

（2）HPLOT　XA，YA　TO　XB，YB(显示直线*AB*)

（3）HPLOT　TO　X，Y（从刚刚显示的某点画线到(X，Y)点）

下面的程序使用指令格式（1）可在屏幕上X＝200，Y＝50处显示一个点。

```
10  REM * DISPLAY A POINT *
20  HGR2
30  HCOLOR = 3
40  HPLOT 200, 50
```

第20行是确定图形显示形式为高分辨率全屏幕图形显示形式。

第30行是选定图形颜色为白色。

第40行中HPLOT后的两个数字，前者是点的X坐标值，后者是Y坐标值，两个坐标值之间要用逗号分隔。坐标值可以用数值（点数），也可以用变量或表达式写出。

使用指令格式（2）可从点$A(XA, YA)$画直线至点$B(XB, YB)$。若设定$XA=200$，$YA=50$，$XB=80$，$YB=140$，用下面的程序可画出直线AB。

```
10  REM * DISPLAY A LINE *
20  HGR2
30  HCOLOR = 3
40  XA = 200:YA = 50:XB = 80:YB = 140
50  HPLOT XA, YA TO XB, YB
```

若要继续画线至点$C(XC, YC)$，…，点$N(XN, YN)$，在给各点坐标变量赋值以后，可把50行写成：

```
50  HPLOT XA, YA TO XB, YB TO XC, YC
    TO…XN, YN
```

指令格式(3)表示从刚刚显示出的某点开始画线到(X, Y)点。所画线的颜色与刚刚显示出的点的颜色相同。

编写程序时应注意X，Y坐标值不得超出屏幕坐标的最大值，也不能是负数。参见图3（a）和图3（b）。

例　编写显示如图 5 所示图形的程序。

此图形很简单，仅由直线构成。只要将计算出的图形各顶点坐标值读出并依次连点画线即可。程序如下：

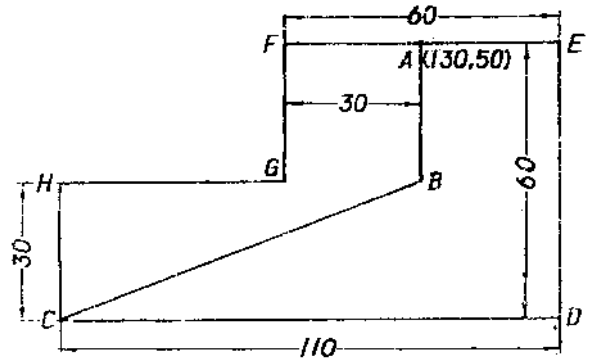


图　5

```
10  REM * DISPLAY A PLANE FIGURE *
20  HGR2
30  HCOLOR = 3
40  HPLOT 130, 50
50  FOR I = 1 TO 8
60  READ X, Y
70  HPLOT TO X, Y
80  NEXT I
90  END
100  DATA 130, 80, 50, 110, 160, 110, 160, 50, 100, 50,
     100, 80, 50, 80, 50, 110
```

由程序可知，画线起点为*A*点，依次画至*B*、*C*、*D*、*E*、*F*、*G*、*H*、*C*各点。

第40行是在*A*处显示一个点。

第50至80行是从A点起依次向B点、C点、…画线（共八条）。若从其它点开始画线，也可仿照此程序另行编程（需修改第40行和100行）。

要注意在上述程序中必须先用40行显示出A点后，才能使70行起作用。若把40行写成HPLOT TO 130，50是不能显示出A点的。

二、基本图形子程序

前面的程序只能显示一个指定的图形，没有通用性。另外，在编程前要算好各点的坐标值，这种编程方法看来简单，但数据量大，容易出错。所以，这样的程序实用价值不大。

在学习组合体的形体分析方法时，一般是把复杂的组合体分为若干个基本几何形体来研究。据此，也可将复杂的图形分解为若干个基本图形，再把不同的基本图形编写出不同的程序。当绘图程序需要时，可对任一基本图形的程序反复调用。可被反复调用的程序称子程序。可被反复调用的基本图形程序，称为基本图形子程序。

下面介绍几个最简单的基本图形子程序的编程方法。

（一）矩形

如图6所示，令矩形的长为L，宽为H，并设点1的坐标为X，Y。该矩形四个顶点的坐标是：

点1坐标（X，Y）

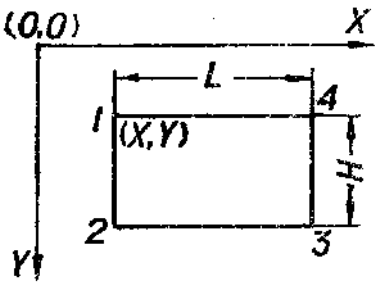


图 6

点 2 坐标（X，$Y+H$）

点 3 坐标（$X+L$，$Y+H$）

点 4 坐标（$X+L$，Y）

显示该矩形的子程序为：

```
500 REM * SUB1 * (X, Y, L, H)
510 HPLOT X, Y TO X, Y + H, TO X +
    L, Y + H TO X + L, Y TO X, Y
520 RETURN
```

（二）左边开口的矩形

变量参数如图 7 所示。其子程序为：

```
600 REM * SUB2 * (X,
    Y, L, H)
610 HPLOT X, Y TO X
    + L, Y TO X+ L,
    Y + H TO X, Y + H
620 RETURN
```

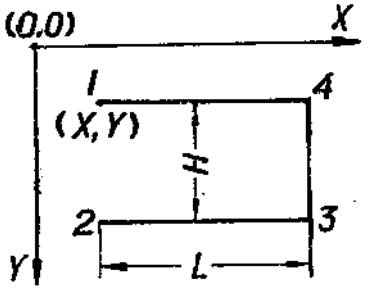


图 7

（三）右边开口的矩形

变量参数如图 8 所示。其子程序为：

```
700 REM * SUB3 * (X,
  Y, L, H)
710 HPLOT X + L, Y TO
    X, Y TO X, Y+ H
    TO X+L, Y+H
720 RETURN
```

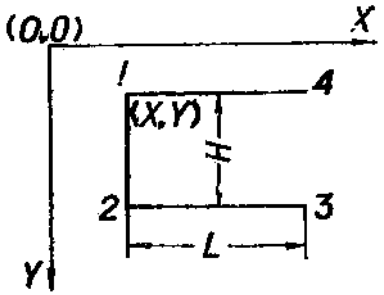


图 8

分析上述子程序的编程方法，可以归纳出以下几点：

1. 首先要选定一个恰当

的点作为画图的定位点（或始画点）；

2．要选取尽可能少的变量参数；

3．写出计算各点坐标的算式；

4．最后编写子程序。

据此，编写显示正多边形的子程序。

（四）正多边形（圆、椭圆）

选正多边形外接圆圆心为画图的定位点，正多边形外接圆半径R、边数N、圆心坐标XC，YC等为变量参数，如图9所示。

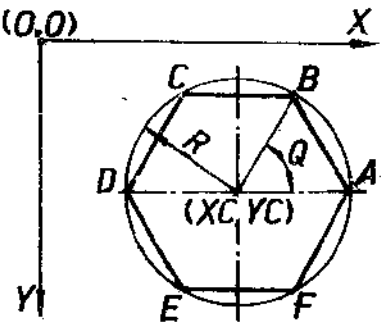


图 9

分析图9可得出正多边形各顶点坐标的表达式如下：

点A坐标　$X=XC+R$

　　$Y=YC$

点B坐标　$X=XC+R*\mathrm{COS}(Q)$

　　$Y=YC-R*SIN(Q)$

点C坐标　$X=XC+R*\mathrm{COS}(2*Q)$

　　$Y=YC-R*SIN(2*Q)$

……

……

点$N-1$坐标　$X=XC+R*\mathrm{COS}((N-1)*Q)$

　　$Y=YC-R*SIN((N-1)*Q)$

点N坐标　$X=XC+R*\mathrm{COS}(N*Q)$

　　$Y=YC-R*SIN(N*Q)$

注：因Y坐标轴向下为正向，故在$R * SIN(Q)$等项前冠以减号。

显示正多边形程序为：（以正六边形为例）

```
10  HGR2:HCOLOR= 3
20  XC=140:YC=95:R=90:N=6:
    Q=6.2832/N
30  HPLOT XC + R, YC
40  FOR I = 1 TO N
50  X=XC + R * COS (I * Q) :Y=
    YC - R * SIN (I * Q)
60  HPLOT TO X, Y
70  NEXT I
80  END
```

第50行算出的X，Y值常常不是整数，但不必为变量值取整。第60行显示图线的指令会自动把X，Y值取整后显示直线。

改变程序中XC、YC、R和N的值，就可显示出不同位置、不同大小、不同边数的正多边形。将80行的END改写为RETURN，就是显示正多边形子程序。

当给定了正多边形的半径后，边数N增加到一定值时（如$N=40$），显示出的正多边形就近似于圆。因而，上述程序也可用于显示圆形。

上述程序稍加修改还可成为显示椭圆的子程序。设RX为椭圆X方向的半径，RY为椭圆Y方向的半径。把上述程序第50行中前后两个变量R分别改写成RX和RY，即成为显示椭圆的程序。修改后的50句为：

```
50  X=XC+RX * COS(I * Q) : Y=YC-RY * SIN(I * Q)
```

画正多边形、圆和椭圆的子程序（由800行开始）为：

```
800  REM * SUB4 * (XC, YC, RX, RY, N)
810  HPLOT XC + RX, YC
820  FOR I = 1 TO N
830  X=XC + RX * COS (I * Q) :Y=
     YC - RY * SIN (I * Q)
840  HPLOT TO X, Y
850  NEXT I
860  HPLOT TO XC + RX, YC
870  RETURN
```

程序中第860行是防止830行因计算误差产生不封闭图形而设立的。

根据程序中变量参数赋值的不同，这个子程序既能显示正多边形，也能显示圆或椭圆。当用其显示正多边形或圆时，变量RX与变量RY的值应相同。当用其显示圆或椭圆时，N值愈大，曲线愈“光滑”，屏幕上显示图形的时间也愈长，一般可取$N=40$。

常用的点划线、虚线、点划线圆、虚线圆等也可编成子程序，这里不再一一介绍。

为了便于调用，现将以上四个子程序的变量参数和起始行号列入表一。

例一　编写一个主程序并调用有关子程序，在屏幕上显示如图10所示的图形（不包括点划线）。

此图由一个矩形和五个圆组成。在编写主程序时，需调用矩形和圆的子程序，并在调用前给所调用子程序中的变量赋值。

表 一

图形名称	变量参数	起始行号
矩形	X, Y, L, H	500
左边开口的矩形	X, Y, L, H	600
右边开口的矩形	X, Y, L, H	700
正多边形、圆、椭圆	XC, YC, RX, RY, N	800

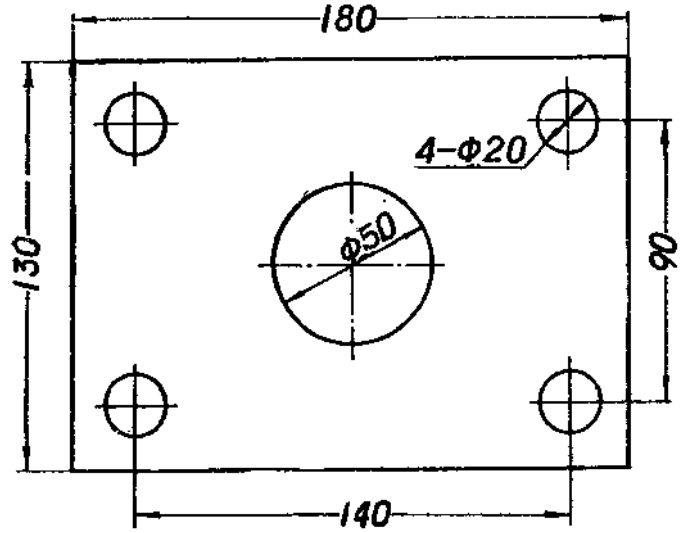


图 10

所编的主程序及被调用的子程序如下：

```
10  REM * MAIN PROGRAM *
20  HGR2:HCOLOR= 3
30  X=50:Y=30:L=180:H=130:
    GOSUB 500
40  N=45:Q=6.2832/N
45  FOR K=1 TO 5: READ XC, YC, RX, RY
50  GOSUB 800
60  NEXT K
65  END
70  DATA 140, 95, 25, 25, 70, 50, 10,
    10, 210, 50, 10, 10, 210, 140, 10, 10,
    70, 140, 10, 10
500 REM * SUB1 *  (X, Y, L, H)
510 HPLOT X, Y TO X, Y + H TO X +
    L, Y + H TO X + L, Y TO X, Y
520 RETURN
800 REM * SUB4 * (XC, YC, RX, RY, N)
810 HPLOT XC + RX, YC
820 FOR I = 1 TO N
830 X=XC + RX * COS (I * Q) :Y=
    YC - RY * SIN (I * Q)
840 HPLOT TO X, Y
850 NEXT I
860 HPLOT TO XC + RX, YC
870 RETURN
```

第30行显示矩形。第40至60行显示一个大圆和四个小圆，其圆心位置和半径大小是利用读数语句（45行后部）赋值的。如大圆圆心设在屏幕中心，坐标为（140，95）。

例二　在屏幕中心显示如图11所示直径为80的四个圆。

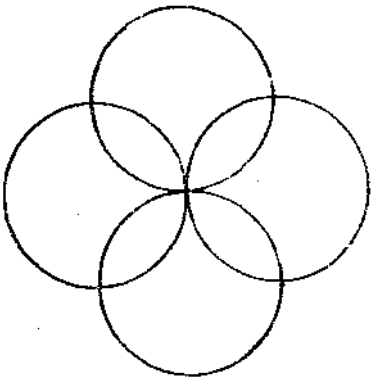


图　11

把四个圆的交点设在屏幕中心（140，95）处，将例一中第30行删掉，第45行中的5改为4，第70行改成：

```
70  DATA 180, 95,
     40, 40, 140, 55,
     40, 40, 100, 95,
     40, 40, 140, 135,
     40,  40
```

其余语句不变，运行修改后的程序，就能按逆时针顺序依次在屏幕上显示四个圆形。

通过上述两例可见：显示图形的程序常常分为主程序和子程序两个部分。显示不同图形时，不需改动子程序，常常只需改变主程序中某些变量的值即可实现。主程序一般要包括下列内容：

选定屏幕图形显示的形式（HGR或HGR2）；

选定图形颜色（HCOLOR = 3）；

给调用的子程序所需变量赋值；

调用子程序（GOSUB 行号）；

结束程序运行（END）。

应当指出：给子程序中所需变量赋值，采用哪种形式的语句都可以，但在调用子程序以前必须先赋值，主程序应有结束程序语句（END）。