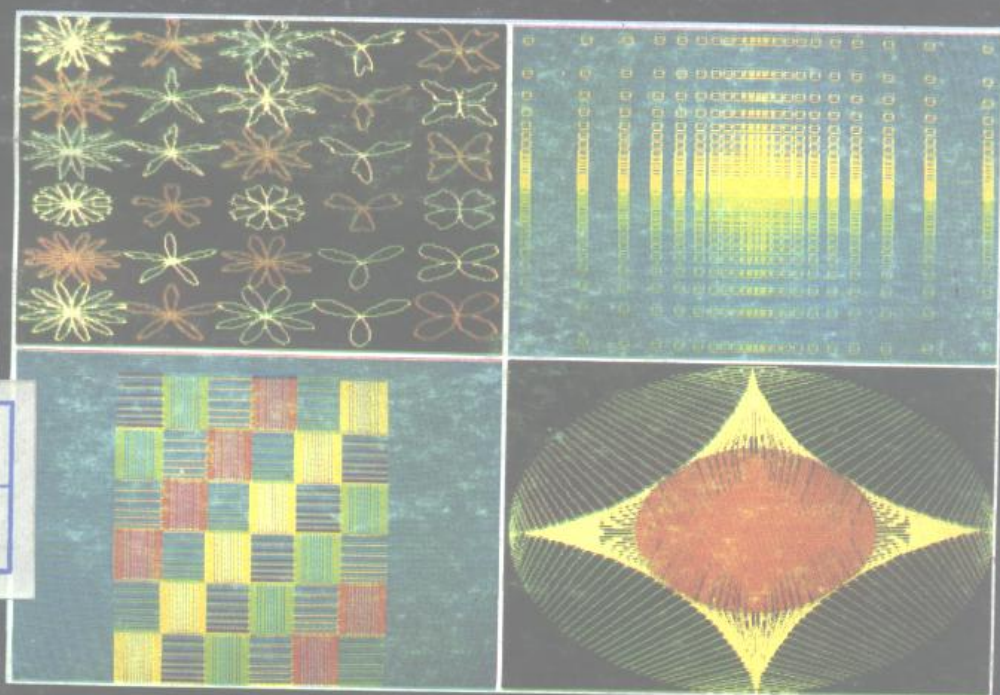


电脑创艺

—BASIC 艺术图案程序设计

李兰友 王俊省 编著
柴宇岚 王强如



电子工业出版社

T1/312
LLY/2

电脑创艺

—BASIC 艺术图案程序设计

李兰友 王俊省 编著
柴宇岚 王强如



电子工业出版社

0021371

内容提要

本书介绍如何使用 BASIC 语言的绘图语句进行平面和立体的艺术绘图程序设计。

全书分：第一部分为 BASIC 语言绘图入门；第二部分为平面图形程序设计；第三部分为三维绘图程序设计。全书给出 77 个绘图程序实例，所用程序均在 IBM PC 机上通过。

为了便于初学者掌握 BASIC 语言的绘图语句及使用技巧，特向读者提供《IBM PC BASIC 绘图辅助教学》软件（一盘）。

全书程序实例软件（二盘）。

JS350/03

电脑创艺

——BASIC 艺术图案程序设计

李兰友 王俊省 编著
柴宇岚 王强如

责任编辑：王惠民

电子工业出版社出版（北京市万寿路）

电子工业出版社发行 各地新华书店经销

艺辉胶印厂印刷

开本：787×1092 毫米 1/32 印张：10.75 字数：221千字

1991 年 3 月第 1 版 1991 年 3 月第 1 次印刷

印数：1~7000 册 定价：5.60 元

ISBN7-5053-1296-0/TP210

编 者 的 话

本书是为已经掌握或初学 BASIC 语言程序设计的读者，介绍如何使用 BASIC 语言的绘图语句设计平面和立体的艺术图案，帮助我们的工艺美术师，广告设计师，服装设计师及美术院校的师生，学会利用计算机进行电脑创艺，绘制出许许多多美丽迷人的图案，让我们的生活充满美的享受。

考虑到我国美术人员中利用电脑绘图尚属初步，为此，书中给出详细的程序清单，所有程序均已在 IBM PC 或 0520 系列微型计算机上通过，读者不妨拿来就用，在实践中增长才干。

为了便于自学，本书配有“绘图辅助教学软盘；及全书例题程序盘片。

参加本书编写及程序调试的还有：韩其睿、张健。

张平合、乐卫平同学开发了“IBM PC BASIC 绘图辅助教学”软件，并编写本书附录部分。

不足之处，望读者赐教。

编者

1989.6.

目 录

第一部分 BASIC 语言绘图入门

第一章 作图语句及屏幕显示	(1)
1.1 屏幕语句	(1)
1.2 屏幕坐标系	(6)
1.3 画点语句 PSET	(9)
1.4 画线语句 LINE	(13)
1.5 连续画线语句	(19)
1.6 画图、椭圆及圆弧	(22)
1.7 着色及阴影	(25)
1.8 字符图案	(32)
1.9 动画	(42)

第二部分 平面图形程度

第二章 图案变换及操作	(53)
2.1 平移变换	(53)
2.2 比例变换	(57)
2.3 旋转变换	(59)
2.4 对称变换和错切变换	(66)
2.5 渐变图案	(74)
2.6 扇形变换	(82)
2.7 环形变换	(84)
2.8 三角形变换	(93)
2.9 三角函数变换	(95)
2.10 球面镜反射变换	(98)
2.11 窗口——视见变换	(102)
2.12 图案的磁盘存取及打印	(109)

2.13 图案的绘图机输出	(114)
第三章 图案程序 60 例	(121)
第三部分 三维绘图程序设计	
第四章 三维绘图入门	(203)
4.1 三维图形的平面显示	(203)
4.2 图形的几何变换	(208)
4.3 透视图	(213)
4.4 三维绘图基本程序包	(216)
第五章 三维绘图程序 17 例	(223)
附录: IBM PC BASIC 绘图辅助教学软件	
(BG1.0)	(313)
参考文献	

第一部分 BASIC 语言

绘图入门

第一章 作图语句及屏幕显示

IBM PC 或 0520 系列微型计算机，配置有彩色图形显示器，具备中分辨率和高分辨率图形的显示能力，可以通过键盘编程，绘制并显示彩色（或黑白）的美术图案。屏幕图案可直接存入磁盘，或在打印机和小型绘图机上绘出。考虑到国内一般微型计算机用户的系统较少配置绘图机，建议读者进行屏幕拍照，在一般屏幕亮度的情况下，采用 21° 的彩色负片，选用光圈 $f8$ ，速度 $1/60$ 秒（最好利用三脚架，便于取景构图），或者变换几挡组合拍摄后，选择出自己的最佳作品，（封面图案就是拍摄的屏幕图案照片）用来作为资料留存、设计作品样稿和印刷用原件都能达到令人满意的程度。

本章结合 BASIC 语言中一些作图的语句，给出实际图案程序，供读者学习参考。

1.1 屏幕语句

在 BASIC 语言控制下，IBM PC 彩色图形显示器有三种工作方式：文本方式；中分辨率图形方式；高分辨率图形方式。

利用 SCREEN 语句，可以选择我们要用的方式。

一. SCREEN 语句

SCREEN 语句格式:

SCREEN 方式值, 色同步, 活动页, 可见页

说明:

方式值的取值为 0、1、2 等数字, 用于选择屏幕工作方式, 规定如下:

数字	方式
0	文本方式
1	中分辨率图形方式
2	高分辨率图形方式

SCREEN 0

选择文本方式;

SCREEN 1

选择为中分辨率图形方式, 而

SCREEN 2

则使屏幕进入高分辨率图形方式。

色同步参量是选用项, 用于说明是否启动色同步信号, 即选择黑白显示还是彩色显示, 取值为 0 或非 0, 一般设定取 0 或 1. 意义如下表:

文本方式	0	黑白显示
	1	彩色显示
中分辨率 图形方式	0	彩色显示
	1	黑色显示

例如:

SCREEN 0, 0 文本方式, 黑白显示;

SCREEN 0, 1 文本方式, 彩色显示。

而

SCREEN 1, 0 中分辨率图形方式, 彩色显示;

色同步参数对高分辨率图形方式不起作用, 因为高分辨率图形方式只使用黑白显示。

活动页和可见页只在文本方式下有效, 活动页是指正在输出的页号; 可见页是指屏幕上显示的页, 取值规定如下:

80 列文本 0~3

40 列文本 0~7

活动页和可见页可取值相同, 也可以不同, 取值不同时用户在看某一页时, 程序可同时写入另一页。

例如:

SCREEN 0, 1, 1, 1, 选择文本方式, 彩色显示,
第 1 页同时做为活动页和可见页;

SCREEN,, 3, 1

选择第 3 页为活动页, 第 1 页为可见页, 方式值和色同步缺省, 表示保持原来的文本方式和显示色彩的选择不变。

二. 宽度语句 WIDTH

WIDTH 语句的格式为:

WIDTH S

说明:

在文本方式下, 参数 S 允许取 40 或 80。

例如:

WIDTH 40

表示每行可显示 40 个字符, 这时文本方式又称为 40 列文本;

WIDTH 80

选择每行显示 80 个字符，又称为 80 列文本。

在图形方式下，语句 WIDTH 40 的作用相当于 SCREEN 1，强行使屏幕进入中分辨率工作方式；而 WIDTH 80 相当于 SCREEN 2 语句，使屏幕工作在高分辨率图形方式。

三. 文本彩色语句 COLOR

在文本方式下，如果启动了色同步，则字符可以用 16 种颜色显示，背景的底色最多能用 8 种与字符不同的颜色。利用 COLOR 语句，可以选择颜色。这时，COLOR 语句的格式为：

COLOR 前景, 背景, 边缘

说明：

前景参数用于选择字符的颜色，背景参数用于选择屏幕底色，边缘参数用于选择文本区域四周的颜色。

16 种颜色的编号如下：

0	黑	8	灰
1	蓝	9	浅蓝
2	绿	10	浅绿
3	青	11	浅青
4	红	12	浅红
5	紫红	13	浅紫红
6	棕	14	黄
7	白	15	高亮度白

前景参数取值范围 0~31，其中取 16~31 时与 0~15 颜色相同，但字符闪烁，例如，前景值 16 表示闪烁的黑字

符,前景 17 表示闪烁的蓝字符等。

背景值只能取 0~7 的数值。

例如: 语句

COLOR 2, 1, 0

说明绿色的字符, 蓝的背景底色和黑色的边缘。而语句

COLOR 0, 4

说明黑色字符、红的背景底色, 边缘色不变。

四. 图形 COLOR 语句

在中分辨率图形方式下启动色同步时, 可以画彩色图形。图形 COLOR 语句的格式为:

COLOR 背景底色, 配色器

说明:

背景底色取值范围在 0~15 之内, 颜色编号与文本 COLOR 语句相同。

配色器取值为 0 或 1 (偶数或奇数), 用于选择配色器 0 或配色器 1。每个配色器配 4 种颜色, 编号为 0, 1, 2, 3, 称为颜色代码。其规定如下:

颜色代码	配色器 0	配色器 1
0	同背景底色	同背景底色
1	绿	青
2	红	紫红
3	棕	白

例如:

COLOR 14, 0

表示背景色是黄的, 并选配色器 0。执行完这个语句后, 颜

色代码 0 是黄, 1 是绿, 2 是红, 3 是棕。

在使用 SCREEN 1 进入中分辨图形方式时, 其初态为: 背景底色是黑色, 配色器是 1 号。此后程序中可以任意使用 COLOR 语句。当 COLOR 语句中的参数缺省时, 缺省某参数则表示保持原来的底色或配色器。

这个语句不适用于高分辨图形方式, 否则会引起语法错误。

五. CLS 语句

CLS 语句用于清除屏幕。在文本方式下, 它把活动页的内容全部清除, 并使光标返回屏幕左上角; 在图形方式下, 它将屏幕全部清为所选择的底色, 并确定下一绘图语句的“最新参考点”为屏幕中心的点。

六. KEY OFF

在绘图中, 利用 KEY OFF 语句清除屏幕下方功能键显示行。

1.2 屏幕坐标系

一. 字符坐标系

由 SCREEN 0 选择文本方式。在文本方式下, 字符显示屏幕如图 1.1。图中虚线框内为图形显示区, 虚线框和边缘之间为边缘区。边缘区不能显示字符。在字符显示区内, 从左往右每行可显示 40 个 (或 80 个) 字符, 垂直方向从上到下可显示 25 行。整个屏幕可显示 1000 (或 2000) 个字符。

取屏幕左上角为坐标原点 (1, 1), 水平方向为 X 轴, 取值 1~40 (或 1~80); 垂直方向为 Y 轴, 从上到下取值为 1~25。这时, 屏幕右下角坐标为 (40, 25) (或 (80,

)))。这种坐标系，称为字符坐标系。

利用字符坐标系，由 LOCATE 语句和 PRINT 语句，可以方便地在屏幕上标注字符或图题说明。例如：

LOCATE 10, 10: PRINT “#” 则在屏幕的第 10 行，第 10 列位置上显示字符“#”。

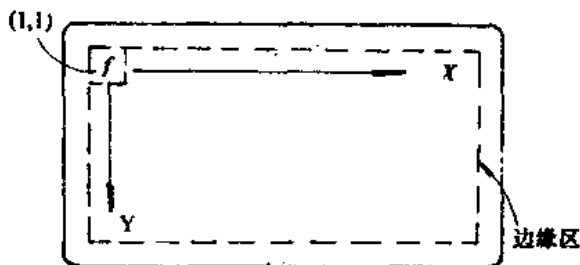


图 1.1

如果行号、列号取值超出屏幕规定取值范围，会出现错误信息。

二. 点坐标系

在图形方式下，屏幕上每一个能够控制的逻辑单元点称为像素，它代表图形的元素。我们称之为点。这时的屏幕坐标系称为点坐标系。

在中分辨率图形方式（由 SCREEN 1 选择）下，屏幕坐标系的原点 (0, 0) 取在屏幕的左上角，水平方向为 X 轴，取值 0~319，垂直方向为 Y 轴，取值为 0~199。这时，屏幕水平方向每行 320 个点，垂直方向 200 个点，整个屏幕可显示 $320 \times 200 = 64000$ 个像素，如图 1.2 所示。这时，屏幕上每个像素的位置可以用它的 X 轴坐标和 Y 轴坐标表示之，记为 (X, Y)。例如，屏幕中心坐标为 (160, 100)，右下角点坐标为 (319, 199)。这种坐标叫做绝对坐

标。每个点的颜色，取决于图形 COLOR 语句中选择的配色器和图形语句中的颜色代码值。

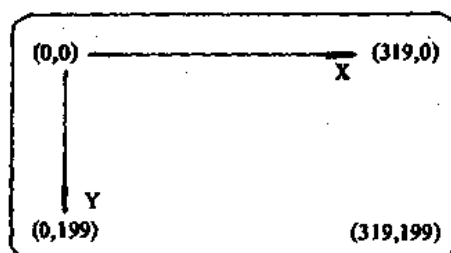


图 1.2

在高分辨率图形方式（由 SCREEN 2 选择）下，点坐标系如图 1.3 所示。X 轴方向取值范围是 0~639，Y 轴方向取值 0~199。整个屏幕可显示 128000 个象素点，每个点只能显示为黑色或白色。

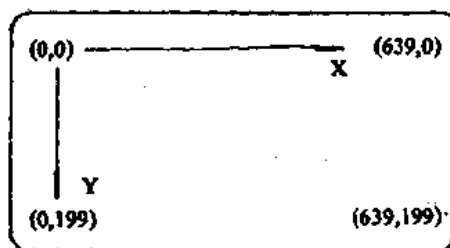


图 1.3

三. 点坐标系和字符坐标系的关系

在图形显示中，往往需要在图形上标注字符。这时要在图形方式下画图，在字符显式方式下标注文字。点坐标系和字符坐标系的对应关系，没有一定的规定。在 640×200 和 80 列文本时，两种坐标系的对应关系如下：

字符坐标	点坐标
(n, m)	$(8(n-1) + 4, 8(m-1) + 3)$
$(1 + (x-4)/8, 1 + (y-3)/8)$	(x, y)

四. 绝对坐标和相对坐标

在图形方式下，绘图语句中点的位置坐标可用两种形式给出。一种是绝对坐标 (X, Y) ，一种是相对坐标，记为 **STEP** (DX, DY) 。这两种坐标都可以用在绘图程序中，但使用时要注意它们的区别：绝对坐标 (X, Y) 的参考点是点坐标系的原点， X, Y 只能取规定范围内的正整数；相对坐标的参考是“画笔”的当前位置， DX, DY 表示相对于“画笔”当前位置在 X 方向和 Y 方向上的偏移量。 DX, DY 可取正数，也可以取负数。例如，“画笔”当前位置点是 $(159, 109)$ ，我们要画出的下一个点位于该点右边 10 列和向上 10 行，这时可用相对坐标 **STEP** $(10, -10)$ 来表示下一个点的位置。这个点的绝对坐标是 $(159+10, 109-10)$ 即 $(169, 99)$ 。

1.3 画点语句 PSET

任何一幅图形都是由许多点组成的，因而画点语句是 IBM PC BASIC 语言绘图中最基本的功能语句。在所有 BASIC 版本中都具有两条用于画点的语句。

一、PSET

格式：

PSET (X, Y) , 颜色代码

说明：

(X, Y) 是欲画点的屏幕坐标。我们可以用绝对坐标，

也可以用相对坐标 STEP (DX, DY)。当坐标取值超出坐标系允许范围时, 语句不起作用。

颜色代码 取值 0~3 (中分辨率) 或 0~1 (高分辨率), 缺省时, 中分辨方式取 3, 高分辨率方式取 1。

例如,

PSET (160, 100), 1

在屏幕中央画一个青点; 这时, 画笔的现行位置在 (160, 100)。当继续执行,

PSET STEP (10,10)

后, 则在屏幕上的 (170, 110) 位置上画一个白点。

例 1-1 画一条正弦曲线和余弦曲线的程序

```
5 'pcl-1
10 SCREEN 1,0:CLS
20 PI=3.14159
30 FOR X=0 TO 2*PI STEP .01
40 S=1-SIN(X): C=1-COS(X)
50 PSET (50*X,99*S),1
60 PSET (50*X,99*C),2
70 NEXT X
80 END
```

程序运行后, 屏幕上显示一条彩色正弦曲线和一条余弦曲线。

二. PRESET (擦点语句)

格式:

PRESET (X,Y), 颜色代码

说明: 这个语句中有关参数规定与 PSET 一样, 也是在指定的屏幕坐标位置处根据颜色代码规定画一个彩色点。与

PSET 不同之处在于，当颜色代码缺省时，PRESET 语句自动选择代码数值。即使用底色画一个点，这相当于在指定位置上将点擦去。

例如，

PSET (250,100), 1

和

PRESET (250,100), 1

都是把绝对坐标 (250, 100) 的点设置为颜色代码 1。而

PSET (250,100)

将点(250,100)自动设置为颜色代码 3，而

PRESET (250,100)

从屏幕上擦去这个点。

实际上，画点语句本身也具有擦点的功能。这时颜色代码取值为 0。

例 1-2 火箭飞升

```
10 'pc 1-2
20 SCREEN 1.0:KEY OFF:CLS
30 LINE (160,190)-(170,200),3,BF
40 FOR HE=178 TO 0 STEP -5
50 C=2:GOSUB 1000
60 C=0:GOSUB 1000
70 NEXT HE
80 END
1000 '
1010 RESTORE : FOR DOT=1 TO 34
1020 READ X,Y
1030 PSET(160+X,HE+Y),C
```

```

1040 NEXT DOT
1050 RETURN
1100 DATA 5,0,4,1,6,1,3,2,7,2
1110 DATA 3,3,7,3,3,4,7,4
1120 DATA 3,5,7,5,3,6,7,6,3,7
1130 DATA 7,7,2,8,8,8,1,9,9,9
1140 DATA 0,10,1,10,2,10,3,10,4,10
1150 DATA 5,10,6,10,7,10,8,10,9,10
1160 DATA 10,10,3,11,4,11,6,11,7,11

```

程序说明:

20 行 设定中分辨率图形方式, 彩色显示, 关闭功能显示行, 清除屏幕;

30 行 画火箭发射架。

50 行 调用子程序以颜色代码 2 画火箭

60 行 调用子程序以颜色代码 0 擦去火箭

40 行 由循环语句控制在不同的位置上画出火箭、擦去火箭, 再在新的位置上画火箭, ……使屏幕上呈现火箭冉冉上升的运动图象。

由点 (0, 0) 到点 (199, 199) 画一条线, 然后由 PRESET 语句擦去的程序:

```

10 SCREEN 1
20 FOR I=0 TO 199
30 PSET (I,I)
40 NEXT I
50 FOR I=199 TO 0 STEP-1
60 PRESET (I,I)
70 NESE I

```

1.4 画线语句 LINE

虽然利用 PSET 和 PRESET 可以画出任何图形，但要设置的点数太多，一个程序要是用 PSET 和 PRESET 每次设置一个点，则设置任何一个实际画面就要花费很多时间，程序也显得冗长。因而，在 IBM PC 高级 BASIC 语言中还提供了一些其他绘图语句，用一个语句可以设置很多点，从而简化程序、提高程序的执行速度。

在大于 2.00 版的 BASICA 语言中，提供了四种格式的画线语句：

格式 1

LINE (X1,Y1)-(X2,Y2), 颜色代码

说明：

(X1,Y1)、(X2,Y2)是画线的始点坐标和终点坐标。可以用绝对坐标，也可以是相对坐标；当(X1, Y1)缺省时，始点为“画笔”现行位置。例如：

```
10 PSET (X1,Y1)
```

```
20 LINE -(X2,Y2),C
```

颜色代码取值 0~3，选择画线的颜色。缺省时取 3（中分辨率图形方式）或 1（高分辨率图形方式）。

例 1-3 在屏幕上每隔 10 点画一条竖线

```
5 'pcl-3
```

```
10 SCREEN 1:CLS
```

```
20 FOR X=0 TO 320 STEP 10
```

```
30 LINE (X,0)-(X,200),2
```

40 NEXT X

50 END

程序执行时，由 10 行进入中分辨率图形方式，并自动选用 1 号配色器。每隔 10 点画一条紫红色竖线。

例 1-4 利用画线语句画一架飞机。如图 1.4 所示。

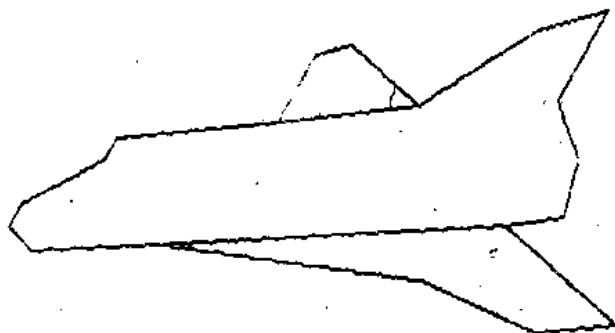


图 1.4

100 'pc 1-4

110 SCREEN 2

120 CLS

130 RESTORE

140 C=5

150 READ X,Y: LINE (X,Y)-(X,Y)

160 FOR I=1 TO 23

170 READ X,Y: LINE -(X,Y)

180 NEXT I

190

200 DATA 420,95,520,60,580,50,540,90,560,115,550,140,500,14

5,600,185,530,190,430,170,200,160,500,145,200,160,90,165,7

0,155,80,145,150,125,160,115,300,105,310,95,330,75,360,70,

420,95,300,105

例 1-5 几何图案

```
100 'pc 1-5
110 SCREEN 2: CLS:KEY OFF
120 XMAX=540:YMAX=190
130 XMIN=100:YMIN=10
140 S=20
150 XS=(XMAX-XMIN)/S:YS=(YMAX-YMIN)/S
160 ' 170 CLS
180 FOR I=0 TO S
190 LINE (XMIN+XS*I,YMAX)-(XMAX,YMAX-YS*I)
200 LINE (XMAX-XS*I,YMIN)-(XMAX,YMAX-YS*I)
210 LINE (XMAX-XS*I,YMIN)-(XMIN,YMIN+YS*I)
220 LINE (XMIN,YMIN+YS*I)-(XMIN+XS*I,YMAX)
230 NEXT
```

程序运行结果如图 1.5.

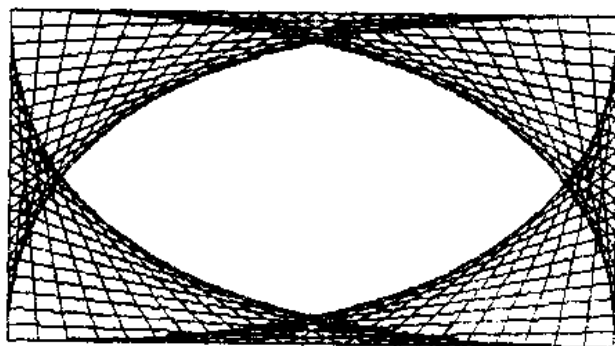


图 1.5

例 1-6 花型图案

```
100 'pcl-6
110 SCREEN 2: CLS
120 PI=3.14159
130 INPUT T
140 D=100
150 FOR A=0 TO 2*PI STEP PI/(20*T)
160 B=D+D/5*SIN(3*T*A)
170 C=B*(1/2+1/2*SIN(T*A))
180 X=320+C*COS(A)*7/5
190 Y=100-C*SIN(A)
200 IF A=0 THEN PSET (X,Y):BX=X:BY=Y ELSE
    LINE-(X,Y)
210 NEXT
220 LINE-(BX,BY)
230 END
```

当取 $T=3$ 时、屏幕
显示图形如图 1.6

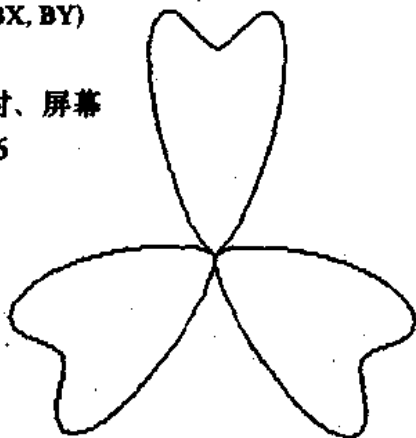


图 1.6

格式 2

LINE (X1,Y1)-(X2,Y2), 颜色代码, B

说明,

其它参数取值与格式 1 相同。B 是一个符号, 表示 (X1, Y1)、(X2, Y2) 为所画矩形左上顶点与右下顶点的坐标。B 虽然是一个符号, 但在画矩形时不可省略。

例 1-7 画一个由小到大变化的矩形

```
5 'PC 1-7
10 CLS: SCREEN 1,0
30 FOR P=7 TO 319 STEP 32
40 C=INT(RND*3)
50 LINE (.25*P, .1*P)-(P, .3*P),C,B
60 NEXT P
70 END
```

程序中由 10 行语句进入中分辨率彩色图形方式, 自动选择 1 号配色器。40 行语句用随机函数选择颜色代码。程序运行后, 屏幕上显示一个由小到大变化的彩色矩形。

格式 3

LINE (X1,Y1)-(X2,Y2), 颜色代码, BF

将格式 2 的符号 B 变更为 BF, 表示画一个充色的矩形。填充色由颜色代码和 COLOR 语句选择的配色器共同决定。BF 是一个符号, 画充色矩形时不可缺省。

例 1-8: 由小到大变化的着色矩形

```
5 'PC 1-8
10 CLS: SCREEN 1,0
30 FOR P=7 TO 319 STEP 32
```

```

40 C=INT(RND*3)
50 LINE(.25*P,.1*P)-(P,.3*P),C,BF
60 NEXT P
70 END

```

这个程序中除 50 行语句的 BF 与前例程序不同外，其余均相同。程序运行后，屏幕上呈现一个由小到大逐渐变化的彩色矩形图案。

格式 4:

LINE (X1,Y1)-(X2,Y2), 颜色代码, B(F), ST

说明:

其他参数意义与前述相同。参数 ST 为 16 进制数，取值范围为 &H0000~&HFFFF。当 16 进制数的每一个二进制位为 0 时，屏幕上不绘出显示点，为 1 时，绘出一个显示点。绘图时从第 1 个位开始操作，到 16 个位为止，然后又从第一位开始，循环操作，因而可以根据需要设计各种装饰线、点划线等。

例 1-9 在屏幕上显示各种点划线的程序

```

10 'pc 1-9
20 SCREEN 2:CLS
30 LINE(0,50)-(639,50)
40 LINE(0,70)-(639,70),,,,&HFF00
50 LINE(0,90)-(639,90),,,,&HF0F0
60 LINE(0,110)-(639,110),,,,&HF99F
70 LINE(0,130)-(639,130),,,,&HE667
80 LINE(0,150)-(639,150),,,,&HE667
90 END

```


例 1-10 点、矩形组成的图案

```
1400 'pc 1-10
1410 SCREEN 1:CLS
1450 FOR I=0 TO 90 STEP 4
1460 X1=160-I:Y1=100-I
1470 X2=160+I:Y2=100+I
1480 LINE (X1,Y1)-(X2,Y2),,B,&HFFFF-I
1490 NEXT I
1500
1510 'dashed lines with style
1520 FOR I=0 TO 180 STEP 4
1530 X1=0:Y1=10+I
1540 X2=319:Y2=200-I
1550 LINE(X1,Y1)-(X2,Y2),2,,&HFFFF-I
1560 NEXT I
```

程序运行时，首先看到用黄色画的矩形方框，从屏幕中心的小方框开始，以越来越大的方框逐渐向外扩大，每个方框线的形式都稍有不同。接下来，可以看到一系列斜穿屏幕的红色线条。每条线具有不同的形式，屏幕上呈现一幅美丽的图案。

1.5 连续画线语句

一幅图往往是由许多线条组成的。因此，程序中需要大量使用 LINE 语句。为了方便地产生图形输出，BASIC 语言中提供了一条连续画线语句。

连续画线语句的格式为：

DRAW 字符串

说明：字符串是由一系列的画图命令组成的，每个画图命令有两部分：一个命令字母及一个数值 n 。画图命令之间相互用分号隔开。

常用字符串画图命令有：

Cn 选择颜色 n 。中分辨率 n 取值 0~3；高分辨率时取 0~1；

Sn 设置比例因子 n 。取值范围 0~255， n 被 4 除是比例因子，其范围为 $1/4 \sim 64$ ；

An 设置画图角度， n 可以是 0, 1, 2, 3，分别对应 0 度、90 度、180 度和 270 度。

Un 向上画线

Dn 向下画线

Ln 向左画线

Rn 向右画线

En 向左上对角线画线

Fn 向右下对角线画线

Gn 向左下对角线画线

Hn 向右上对角线画线

M_{x,y} 从现行位置向 (X, Y) 画线。若 X, Y 前标有正负号，则表示以现行位置为起点，按相对坐标画线。

An 旋转 n 角度值，其取值范围在 -360 至 360 度之间。若 n 为正数，则以反时针方向旋转； n 值为负数，顺时针方向旋转。 n 值超出范围，产生错误信息。

X 字符串。命令字符串中包含一个或 n 个子命令字符串。

P_{a,b} 填色，P 表示填入颜色代码，取值 0~3；b 为设定边界颜色，设定方式与 P 相同。

在上述字符串命令中，U、D、L、R、E、F、G、H 及 M 命令前可加前缀 B 或 N，前缀字母意义为：

B 移动但不画线。

N 移动，画图完成后返回原来位置。

例 1-11 画一个逐渐增大的三角形图案的程序

```
5 'pcl-11
10 SCREEN 1:CLS
20 TR($)="ufl"
30 DRAW "bm210,10;c3;a0"
40 FOR K=1 TO 18
50 SZ=4*K
60 DRAW "s=sz"
70 DRAW "bg+TR $"
80 NEXT K
90 END
```

例 1-12 使用 DRAW 语句画一只帆船

```
10 'pcl-12
20 SCREEN 1:COLOR 0,0:CLS
30 DRAW "bm60,140 r200 g30 l140 h30"
40 DEAW "bm150,130 r110 h110 d120"
50 DEAW "bm150,130l80 c80"
60 IF INKEY $="" THEN 60
70 END
```

例 1-13 画一个不断增大的五角星图案

```
5 'pcl-13
10 SCREEN 2:CLS
20 DRAW "bm300,25"
```

```

30 STAR $ = "m+7,17;m-17,-12;m+20,0;m-17,12m+7,-17"
40 FOR SCALE=1 TO 40 STEP 2
50 DRAW "c1;s=scale;bm-2,0;x star $;"
60 NEXT SCALE
70 END

```

程序运行结果如图 1.7

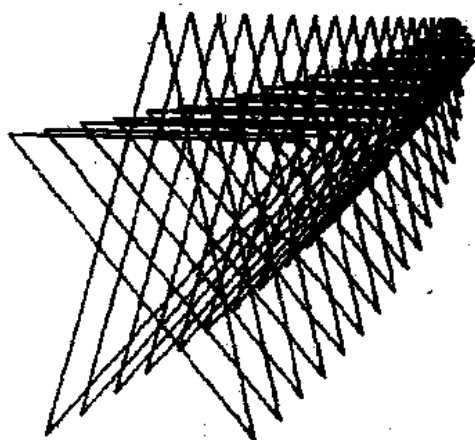


图 1.7

1.6 画圆、椭圆及圆弧

在 BASIC 语言中，提供一条 CIRCLE 语句用于直接画圆、画椭圆及圆弧。

画圆语句格式为：

CIRCLE (X,Y), 半径, 颜色代码

(X, Y)为圆心坐标。例如，

CLRCLE (160, 100), 50 表示以 (160, 100) 为圆心，以 50 为半径画一个圆；而

CLRCLE (160,100), 50, 2

表示画一个与前述不同颜色的圆。

例 1-14 画同心圆图案

```
5 'pcl-14
7 SCREEN 1,0:CLS
10 FOR I=1 TO 300 STEP 10
20 CIRCLE (150,100),I,2
30 NEXT I
40 END
```

例 1-15 画向右连续移动圆的图案

```
1 'pcl-15
5 SCREEN 2:CLS
10 FOR I=1 TO 640 STEP 20
20 CIRCLE (I,100),50
30 NEXT I
40 END
```

画图弧的格式如下:

CIRCLE (X,Y),半径,颜色代码,起始角,终止角,

说明:

起始角和终止角的角度要用弧度来表示。取值范围为 $-2\pi \sim 2\pi$ ，以逆时针方向进行度量。当角度为负值时，表示圆弧的端点应与圆心相连。当这两个参数缺省时，缺省值为0。例如语句:

CIRCLE (160,100),50,2,1.57,4.71

能画出一个半圆；又例如

CIRCLR (160,100),50,2,-.785,-1.57

能画出一个像一块馅饼样子的图形，它的大小是馅饼的 1/8。

决定是画圆还是画椭圆的参数是纵横比。这时，画图语句的格式为：

CIRCLE (X,Y), 半径, 颜色代码,... 纵横比

当纵横比参数缺省时，画圆；当设定纵横比参数时，画椭圆。例如。

CIRCLE (160,100),75,,,1 则画一个椭圆。

例 1-16 画圆弧程序例

```
10 'pci-15
20 SCREEN 2:CLS
30 FOR I=30 TO 180 STEP 30
40 CIRCLE (I / 30 * 110,55), 50,, -.00001, -I * 3.14 / 180
50 NEXT I
60 FOR I=210 TO 330 STEP 30
70 CIRCLE (I / 30 * 110-660,110), 50,, -.00001, -I * 3.14 / 180
80 NEXT
90 END
```

例 1-17. 使用 CIRCLE 语句画椭圆

```
10 'pci-17
20 SCREEN 2:CLS
30 FOR I=1 TO 100 STEP 5
40 CIRCLE (320,100),I,,,I / 100
50 NEXT I
60 END
```

程序运行后，屏幕上显示出由一组椭圆组成的图案。

1.7 着色及阴影

一. 图形着色

先输入程序

例 1-18

```
5  'pc1-18
10 SCREEN 1,0:CLS
20 X=RND*319:Y=RND*40
30 RADIUS=RND*40
40 CIRCLE(X,Y),RADIUS,2
50 GOTO 20
```

这个程序随机地取屏幕上的一点为圆心，并以任意的一个小于 40 的数为半径画圆。圆的边线为紫红色，屏幕底色为黑色。程序运行后将无限循环下去。只有按下 Ctrl-Break 键才会停止运行。如图 1.8 所示。

如果将圆内着上各种不同的颜色，那么这个程序就更加有趣了。将程序 PC1-18 改为：

例 1-19

```
5  'pc1-19
10 SCREEN 1,0:CLS
20 X=RND*319:Y=RND*199
25 CLR=INT(RND*3)
30 RADIUS=RND*40
40 CIRCLE(X,Y),RADIUS,2
45 PAINT(X,Y),CLR,2
50 GOTO 20
```

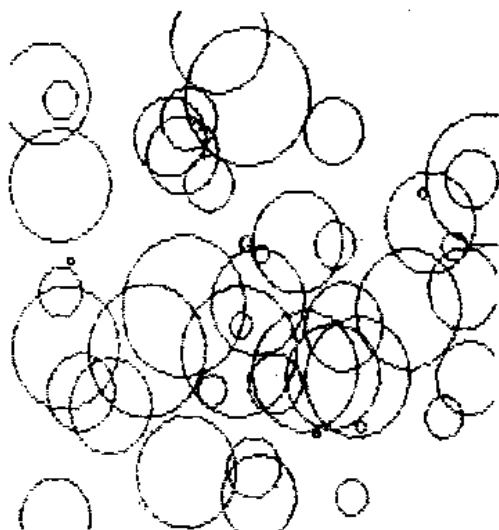


图 1.8

执行这个程序后，你会看到一幅边界线为紫红色，而圆内（确切讲为封闭圆内）涂满了某一种（紫红、白色或青色）颜色的五彩“气泡”（见封面图）。

比较上述两个程序例，可以看出，后面一个程序比前面一个程序多了两个语句：

```
25 CLR=INT(RND*4)
```

```
45 PAINT(X,Y),CLR,2
```

其中 25 行语句是随机选用颜色的语句；变量 CLR 可以取 1 号配色器内 4 种颜色代码，而语句 45 是一个新语句，称为涂色语句。

PAINT 语句格式

PAINT (X,Y),填充颜色, 边线颜色

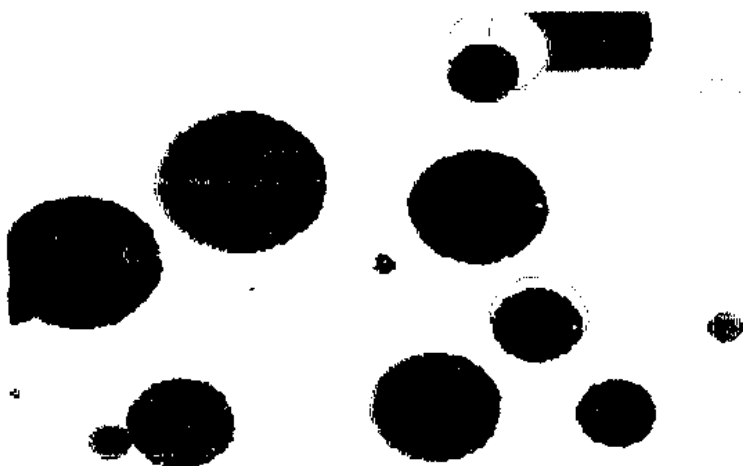


图 1.9

参数(X,Y)是待涂色区域内一点的坐标值，涂色从此点开始。这个坐标值可以取绝对坐标，也可以是相对坐标。

填充颜色代码取值 0~3。缺省时取值为 3。

边线颜色码可在 0~3 内取值，表示图案边界色。

PAINT 语句的功能，是用所选择的填充颜色去涂满屏幕上由边界线所围成的一个闭合区域。涂色开始点必须画在图形里面。如果这个点已有由边线色码所选择的颜色，则 PAINT 语句无效。

如果点 (X, Y) 不是在一个封闭的区域内，“涂色”将“漏出”而充满整个屏幕。

例如：

```
5 SCREEN 1
```

```
10 LINE (0,0)-(100,150),2,B
```

20 PAINT (50,50),1,2

则由 20 行语句，由青色充满矩形，矩形边框线为紫红色。这样，PAINT 语句首先要在封闭图形内，例如矩形内选择一点，然后指明涂上什么颜色，最后给出一个边界色，以说明它应该在什么地方停止涂色。

因此，在使用 PAINT 语句时，要特别注意一定要从一个封闭的区域内部开始涂起，其次要检查有没有“漏洞”，否则会漏出到整个屏幕。

例 1-20

```
5  'pci-20
10 SCREEN 1,0:CLS:KEY OFF
20 LINE-(319,150),1
30 LINE-(319,199),1
40 LINE-(160,100),1
50 PAINT(318,151),2,1
60 END
```

执行这个程序，则在屏幕上显示一个边线为青色、内部涂满紫红色的三角形。

如果将程序中 50 行语句改为：

```
50 PAINT (318,199),2,1
```

涂色开始点(318,199)不在闭合三角形内。再执行这个程序时，则会看到，除青色三角形内外，紫红色充满整个屏幕。

例 1-21 画一个带字符图案的涂色立方体程序

```
1140 'PC1-21
1150 'block
1170 T$ = "block"
1180 SCREEN 1,0: CLS
```

1190 COLOR 0,0
1250 PSET (100,150),1
1260 LINE-(180,80),1,BF
1290 PSET (100,80),2
1300 LINE-(180,80),2
1310 LINE-(210,50),2
1320 LINE-(130,50),2
1330 LINE-(100,80),2
1340 PAINT (120,70),2
1370 PSET (180,150),3
1380 LINE-(210,120),3
1390 LINE-(210,50),3
1400 LINE-(180,80),3
1410 LINE-(180,150),3
1420 PAINT(200,100),3
1560 PSET (110,140),0
1570 LINE-(135,90),3
1580 LINE-(145,90),0
1590 LINE-(170,140),0
1600 LINE-(160,140),0
1610 LINE-(155,130),0
1620 LINE-(125,130),0
1630 LINE-(120,140),0
1640 LINE-(110,140),0
1650 PSET (130,120),0
1660 LINE-(150,120),0
1670 LINE-(140,100),0

```

1680 LINE-(130,120),0
1690 PAINT STEP(-1,-1),2,0
1700 END

```

程序运行后，在屏幕上呈现一个彩色立方体，立方体正面绘有一个字符 A 的图案。

二、混合色和涂阴影

利用两个配色器提供的七种颜色，由程序可以调配出许多种过渡色。例如，用某一种颜色画一条直线，再用另一种颜色在旁边画另一条直线，就会出现第三种颜色。用这种方法，可以组合出各种不同的颜色。

例 1-22. 调色程序

```

5' pc1-22
10 SCREEN 1,0:CLS
20 COLOR C1
30 FOR A=0 TO 3
40 FOR B=0 TO 3
50 FOR X=100 TO 210 STEP 2
60 LINE (X,75)-(X,175),A
70 LINE (X+1,75)-(X+1,175),B
80 NEXT X,B,A
90 FOR A=0 TO 3
100 FOR B=0 TO 3
110 FOR Y=75 TO 175 STEP 2
120 LINE (100,Y)-(210,Y),A
130 LINE (100,Y+1)-(210,Y+1); B
140 NEXT Y,B,A

```

这个程序实际上是两个程序。30~90 行进行纵向直线

的颜色调配。其中颜色 A 与 B 由 (0.0) 组合一直变到 (3.3) 组合。100~160 行语句的作法相同，只不过是画横向直线。

为了取得更好的画面效果，除了着色以外，有时还可以在画面的某些区域涂上深浅不同的有明暗变化的阴影。画阴影和画纹理有多种方法，可以视实际需要而定，但大多数都只是着色技术的一些变化。例如，在着色算法中，每隔一条或两条线着色一次，或者使间隔逐渐增加或减少，都能得到明暗有变化的阴影。在边线所规定的区域内，也可以用画点的方法产生阴影，它们的间距可以是有规律的变化，也可以是随机变化，均能达到预期效果。如果使用不同颜色的点，则效果更佳。

例 1-23 涂阴影程序

```
5' pc1-23
10 SCREEN 1,0:COLOR 4,0:CLS
20 COLUMN=.5
25 READ X1,X2,Y1,Y2,C,C1
26 DATA 10,100,10,100,3,3
30 LINE (X1,Y1)-(X2,Y2),C,B
40 COLUMN=1.2 * COLUMN
50 X=X1+COLUMN:Y=Y2-COLUMN
60 IF X>X2 THEN 80
70 LINE (X,Y1)-(X,Y2),C1
80 IF Y<Y1 THEN 110
90 LINE (X1,Y)-(X2,Y),C1
100 GOTO 40
110 END
```

1.8 字符图案

在文本方式下，由 256 个字符及字符组合，可以在屏幕上显示各种字符图形。

一、简单字符图形

例 1-24 由 PRINT 语句在屏幕上绘一幅图形

[illegible]

程序运行后，在屏幕上显示出一幅由字符 0 组成的图案。

例 1-25 画出啄木鸟外型图

10 'SAMPLE 2: Draw Woodstock

```

20 PRINT "      ,  -"
30 PRINT "      ,  (  -"
40 PRINT "      ,  (  - - -"
50 PRINT "      -  - - - -"
60 PRINT "      -(  - - - - -"
70 PRINT "      , = = = = = *  - - = "
80 PRINT "      (  - - - - -"
90 PRINT "      ,  -"
100 PRINT "      - - - - -      = = = = "
110 PRINT "      -      = = -"
120 PRINT "      -  - - - - = "
130 PRINT "      -      = -"
140 PRINT "      ( , "
150 PRINT "      ( , , "
160 PRINT "      , , , "
170 PRINT "      , , , , "
180 PRINT "      , , , , , "
190 PRINT "      , , , , , "
200 PRINT "      , , , , , "
210 PRINT "      = - - - - - * = = = = -"
220 PRINT "      1  = = = = = = = = "
230 PRINT "      ((((((( -"
240 PRINT "      ,("

```

程序运行结果如图 1.10。

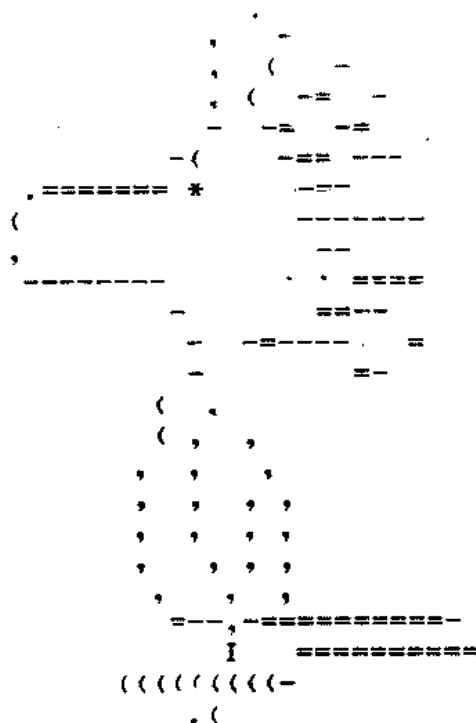


图 1.10

从上面两个例子中看到，这种画图方法很简单。首先把原图描绘在编码表格纸上，然后使用 PRINT 语句，将图形显示在屏幕上。

例 1-26 花纹织物图案程序

```

10 'pcl-26
20 READ X
30 IF X=-1 THEN 999
40 IF X=1 THEN 100
50 IF X=2 THEN 120
60 IF X=3 THEN 140
70 IF X=4 THEN 160
80 IF X=5 THEN 180

```



```

90 GOTO 20
100 PRINT " * * * * * — * * * — * * * * * "
110 GOTO 20
120 PRINT " * * * — * * * * — * * * — * * * "
130 GOTO 20
140 PRINT " * * * * * * * * * * * * * * * * "
150 GOTO 20
160 PRINT " * — * — * — * — * — * — * — * — * "
170 GOTO 20
180 PRINT " * * * — * * * — * * * — * * * "
190 GOTO 20
900 DATA 3, 4, 4, 5, 1, 5, 1, 5, 1, 5, 2, 2
910 DATA 2, 5, 1, 5, 1, 5, 1, 5, 4, 4, 3, -1
999 END

```

程序运行后，屏幕上呈现一幅美丽的小花纹织物图案。

二、利用 TAB 函数在屏幕上绘图

各种版本 BASIC 都提供了 TAB 函数，用来调整图形在屏幕上的位置。利用 PRINT 语句和 TAB 函数，可以在屏幕上绘制各式各样的字符图形。

例 1-27 画正弦函数的图形

```

5 'pcl-27
20 FOR X=0 TO 6.28 STEP .25
30 Y=SIN(X)*30
40 PRINT TAB(Y+30); "*"
50 NEXT X
60 END

```

程序中，Y 表示横向，X 表示纵向，当 X 值变化时，

则以 (*) 号表示 Y 位置。若将 PRINT 指令改为 LPRINT, 就可以由打印机打印出图形。

程序中每隔 0.25 弧度取一个 X 值, 增加 X 的差距, 可以缩短正弦函数的图形。这个程序画出了 $0 \sim 2\pi$ 弧度的正弦图形。

例 1-28 编写一个用不同字符同时显示 $a = \sin x$, $b = \sin 2x$, $c = a+b$ 的程序。

```
10 'pcl-28
20 FOR X=.1 TO 6.28 STEP=.2
30 S=SIN(X): S2=(SIN(2 * X)): SS=S+S2
40 A=20 * S+32: B=20 * S2+32: C=20 * SS+32
50 IF A <= B AND A <= C THEN PRINT TAB(A) ". ";
   ELSE GOTO 70
60 IF B <= C THEN PRINT TAB(B) "+" TAB(C) "*" ELSE
   PRINT TAB(C) "*" TAB(B) "+"
70 IF B <= A AND B <= C THEN PRINT TAB(B) "+" ;
   ELSE GOTO 90
80 IF A <= C THEN PRINT TAB(A) ". " TAB(C) "*" ELSE
   PRINT TAB(C) "*" TAB(A) ". "
90 IF C <= A AND C <= B THEN PRINT TAB(C)
   "*"; ELSE GOTO 110
100 IF A <= B THEN PRINT TAB(A) ". " TAB(B) "+" ELSE
   PRINT TAB(B) "+" TAB(A) ". "
110 NEXT X
120 END
```

这是一个很有趣的尝试。当取

$$y = \sin x + \sin 2x + \sin 3x + \sin 4x + \sin 5x$$

时，程序运行结果将是一个锯齿波，如果取

$$y = \sin x + \sin 3x + \sin 5x + \sin 7x + \sin 9x$$

其执行结果是一个近似方波。

三、字符画图法

IBM PC 提供了 256 个字符可供选用，其中前 128 个字符是标准的 ASCII 字元，并写在键盘上，后 128 个字符存贮在系统的只读存储器 ROM 内，可以用来绘制各种图表和图案。

例 1-29 利用 IBM PC 字符建立一幅字符影幕的程序

```
10 'oci-29
50 DIM C$(64):CLS:K=0
60 FOR I=1 TO 4
70 READ B,E
80 FOR J=B TO E:K=K+1:C$(K)=CHR$(J):NEXT J
90 NEXT I
100 FOR I=1 TO 1024
110 J=1+INT(80*RND):K=1+INT(24*RND):LOCATE K,
    J
120 L=1+INT(64*RND):PRINT C$(L);
130 NEXT I
140 DATA 1,6,14,22,176,223,254,254
150 END
```

例 1-30 画横向条形统计图通用程序

```
20 'pci-30
30 SCREEN 0:CLS
50 WIDTH 80
```

```

60 '
70 REM * * * data input * * *
80 DIM D(12)
90 FOR I=1 TO 12
100 PRINT I
110 INPUT D(I)
120 NEXT I
130 '
140 REM * * * max data * * *
150 MAX=D(1)
160 FOR J=2 TO 12
170 X=D(J)
180 IF MAX < X THEN 200
190 MAX=X
200 NEXT J
210 '
220 REM * * * bouhyouzy * * *
230 FOR I=1 TO 12
240 PRINT "no.": I; TAB(10); "I" ;
250 FOR R=1 TO D(I) STEP MAX/50
260 X=X+1
270 PRINT "* ";
280 NEXT R
290 PRINT TAB(65); USING "#####"; D(I)
300 NEXT I
310 PRINT TAB(10); "+";
320 FOR I=1 TO 50

```

```

330 IF I / 10 = INT(I / 10) THEN PRINT "+"; ELSE PRINT
    "-";
340 NEXT I
350 PRINT
360 X = 10
370 FOR I = 1 TO 5
380 X = X + 10
390 PRINT TAB(X); MAX / 5 * I;
400 NEXT I
410 END

```

程序运行后，按 RUN 键后，由键盘顺序输入数据（本程序最多输入 12 个数据），再按回车键，屏幕上将显示出统计图（图 1.11）。

例 1-31 画字符纵向条形图程序

输入 12 个数据后，屏幕上呈现字符纵向条形图。

```

20 'pcl-31
30 SCREEN 0: CLS
50 WIDTH 80
60 '
70 REM * * * data input * * *
80 DIM D(15)
90 FOR I = 1 TO 12
100 PRINT I
110 INPUT D(I)
120 NEXT I
130 '
140 REM * * * max data * * *

```

4	BE	0 或 45
5	AD	1~5
6	A	0.1~1.5 (最小最长 0.1)
7	D	0.5~3 (最小最长 0.5)
8	SF	0.2~1 (最小最长 0.1)

程序运行结果例如图 3.67。

```

100 'PC3-57
110 SE1 = 2
130 SCREEN 2:WIDTH 80
140 PI = 3.14159
170 INPUT "un = 1-6"; UN
180 UV = 360 / UN:K = UV / 2:HF1 = 180 - UV:HF2 = 180 - K
190 SS = UV / 100
200 INPUT "p0 = 1-4"; P0
210 IF P0 = 1 THEN BX = HF2:BY = HF2
220 IF P0 = 2 THEN BX = HF1:BY = HF1
230 IF P0 = 3 THEN BX = HF2:BY = HF1
240 IF P0 = 4 THEN BX = HF1:BY = HF2
310 INPUT "ell = 1-5"; EL1
320 IF BL1 = 5 THEN ST2 = 50 ELSE ST2 = 2
330 INPUT "be = 0 or 45"; BE
340 IF BE = 45 THEN BE = PI / 4 ELSE BE = 0
380 INPUT "ad = 1-5"; AD
390 IF AD = 1 THEN ED = 0:ST = 1
400 IF AD = 2 THEN ED = PI / 4:ST = PI / 4
410 IF AD = 3 THEN ED = PI / 33:ST = PI / 6
420 IF AD = 4 THEN ED = 3 * PI / 8:ST = PI / 8
430 IF AD = 5 THEN ED = 2 * PI / 5:ST = PI / 10
440 DIM X(52 * AD * 4), Y(51 * AD * 4), PT(40)
450 IF EL1 = 5 THEN GG = -1:GOTO 490 ELSE GG = 1
460 INPUT "a = (0.1-1.5)"; A:CLS
470 INPUT "d = 0.5-2.5"; D
480 IF D < 2 THEN ST2 = 5

```

```

150 MAX=D(1)
160 FOR J=2 TO 12
170 X=D(J)
180 IF MAX>=X THEN 200
190 MAX=X
200 NEXT J
210 '
220 Y=10
230 FOR X=3 TO 14
240 LOCATE X,Y:PRINT CHR$(179)
250 NEXT X
260 Y=10:X=15
270 LOCATE X,Y:PRINT "+";
280 X=15
290 FOR Y=11 TO 70
300 LOCATE X,Y:PRINT "-";
420 NEXT Y
430 Y=11:X=15
440 FOR J=1 TO 12
450 Y=Y+5
460 LOCATE X,Y:PRINT "+";
470 NEXT J
480 Y=10:X=15
490 FOR J=1 TO 4
500 X=X-3
510 LOCATE X,Y:PRINT "+"
520 NEXT J

```

```

530 Y=3: X=15
540 FOR J=1 TO 3
550 X=X-3
560 LOCATE X,Y: PRINT USING "#####"; MAX
    / 4 * J
570 NEXT J
580 '
600 Y=7
610 FOR I=1 TO 12
620 Y=Y+5: X=15
630 LOCATE X+1, Y-1: PRINT I; "M"
640 FOR R=1 TO D(I) STEP MAX / 12
650 X=X-1
660 LOCATE X,Y: PRINT " * * * * "
670 NEXT R
680 LOCATE I, Y-1: PRINT USING "#####"; D(I)
690 NEXT I
700 LOCATE 18, 45: INPUT "next ok"; A

```

1.9 动画

通常把屏幕上显示的图象或图形一部分，按照一定的规则在屏幕上移动而产生的画面效果叫做动画。

一、简单动画

例 1-32 小狗在画面上跑动

```

80 'pc1-32
90 SCREEN 0: WIDTH 80: CLS
100 FOR X=1 TO 70 STEP 4

```



```

110 LOCATE 11, X: PRINT "...$ "
120 LOCATE 12, X: PRINT " / "; CHR$(219); CHR$(
    (219); CHR$(219); CHR$(219); CHR$(219)
130 LOCATE 13, X: PRINT " "; " / "; CHR$(92); " / ";
    CHR$(92)
140 FOR I=1 TO 100: NEXT I
150 LOCATE 11, X: PRINT " "
160 LOCATE 12, X: PRINT " "
170 LOCATE 13, X: PRINT " "
180 LOCATE 11, X+2: PRINT CHR$(92); " "; " ";
    " "; " $))
190 LOCATE 11, X+2: PRINT " "; CHR$(219); CHR$(
    (219); CHR$(219); CHR$(219); CHR$(219)
200 LOCATE 13, X+2: PRINT " "; " / "; CHR$(92);
    " / "; CHR$(92)
210 FOR I=1 TO 100: NEXT I
220 LOCATE 11, X+2: PRINT " "
230 LOCATE 12, X+2: PRINT " "
240 LOCATE 13, X+2: PRINT " "
250 NEXT X
260 END

```

程序中由 110~130 行语句用字符画出一幅小狗的图案，而后由 150~170 行语句将小狗图案擦去；接着由 180~200 行语句在屏幕的新的位置上显示一幅小狗的图案，再由 220~240 行语句将图案擦去，……，如此往复循环下去，我们在屏幕上就会看到一只小狗在跑动的画面。

例 1-33 高尔夫球手走动

```

80 'pc1-33
90 SCREEN 0: WIDTH 40: CLS
100 C$(1)=" ■ "
110 C$(2)="|/■>"
120 C$(3)="J >\ "
130 D$(1)=" ■"
140 D$(2)=" < ■\|"
150 D$(3)=" XL"
190 SW=0
200 FOR X=5 TO 30
210 IF SW=1 THEN 300
220 FOR J=1 TO 100: NEXT J
230 FOR Y=19 TO 22
240 LOCATE Y,X-1: PRINT "
250 NEXT Y
260 FOR Y=20 TO 22
270 LOCATE Y,X: PRINT C$(Y-19)
280 NEXT Y
290 SW=1: GOTO 390
300 '
310 FOR J=1 TO 100: NEXT J
320 FOR Y=19 TO 22
330 LOCATE Y,X-1: PRINT "
340 NEXT Y
350 FOR Y=20 TO 22
360 LOCATE Y,X: PRINT D$(Y-19)
370 NEXT Y

```

```
380 SW=0
390 NEXT X
400 END
```

例 1-34 秒表

```
5 'pcl-34
10 SCREEN 2:CLS
20 DIM X(360),Y(360)
30 R=106:S=-1:GOSUB 140
40 R=100:S=-6:GOSUB 140
50 R=94:S=-6:SW=1:GOSUB 140
60 FOR I=1 TO 60
70 LINE (320,100)-(X(I),Y(I))
80 IF (I=1) AND (J=11) THEN BEEP
90 IF I>57 THEN BEEP:FOR J=1 TO 10:NEXT J:BEEP
100 P$=TIME$
110 IF P$=TIME$ GOTO 110
120 LINE (320,100)-(X(I),Y(I)),0
130 NEXT I:GOTO 60
140 J=0:FOR I=450 TO 91 STEP S
150 J=J+1
160 X(J)=INT(R * COS(I * 3.142 / 180)+320)
170 Y(J)=INT(100-R / 2 * SIN(I * 3.142 / 180))
180 IF SW=0 THEN PSET (X(J),Y(J))
190 NEXT I
200 RETURN
```

每秒秒针跳动 1 次，到 57 秒时开始发出声音，鸣叫四次。

例 1-35 滑翔机从远处飞来

利用 DRAW 语句的 A 命令和 S 命令，可以画出一个运动的物体。

```
5  'pcl-35
10 SCREEN 1,0:CLS:KEY OFF
30 FOR SCA=1 TO 8 STEP .3
40 S=SCA*2:CLS
50 DRAW "BM160,100 A=AN; S=S; BM-4,+2"
60 DRAW " C3 U4 M+2,-2R4 M+2,+2 D4 M-2,+2 L4
    M-2,-2"
70 DRAW " C1 BU1 M-17,-1 M+17,-1 BR8 M+17,+1
    M-17,+1"
80 DRAW :C3 BU3 L8 BU1 BR1 R6 BL3 D1"
90 DRAW " C2 BU2 BL1 M+1,-5 M+1,+5"
100 FOR I=1 TO 300:NEXT I
110 NEXT SCA
120 NEXT AN
130 END
```

50~90 行，用三种颜色画飞机，首先由 50 行中的“BM160, 100”指定从屏幕中心点开始起画，由循环语句分别控制 A 和 S 这两个变量，线性地增大飞机的尺寸，程序运行时，一架飞机在屏幕上迎面飞来，越来越大，最后布满整个屏幕，然后旋转方向，反复飞行。

二、快速动画

当动画对象图案复杂时，则每次花费在擦去图形和画画图形上的时间要增加，动画对象越复杂，增加的时间越多。如果运动轨迹的计算又复杂时，则动画速度就会很慢，使动

画效果不太理想。

为了实现快速动画，BASIC 语言在文本方式下提供了分页动画功能，并有两个专门用于图形方式的语句 GET 和 PUT。

1. 分页动画

在文本方式下，为加快动画速度，可以使用多个显示页同时进行操作。

例 1-36 火箭飞升

```
5 'pcl-36
10 SCREEN 0,1: WIDTH 40: KEY OFF
20 CLS
30 C=18: COLOR 12
40 FOR SCR=7 TO 0 STEP -1
50 R=2*(1+SCR)
60 GOSUB 300
65 SCREEN,,SCR
70 NEXT SCR
80 SCREEN,,0,0
90 FOR TR=1 TO 10
100 FOR SCR=7 TO 0 STEP -1
110 SCREEN,,SCR
120 FOR J=0 TO 200: NEXT J
130 NEXT SCR
140 NEXT TR
150 SCREEN 0,1,0,0
160 END
300 '
```

```

310 LOCATE R+1, C: PRINT " "; CHR$(65); " "
320 LOCATE R+2, C: PRINT " "; CHR$(177); " "
330 LOCATE R+3, C: PRINT " "; CHR$(219); " "
340 LOCATE R+4, C: PRINT CHR$(174); " "; CHR$(175)
350 LOCATE R+5, C: PRINT " "; CHR$(186); " "
360 RETURN

```

这是一个火箭飞升的动画程序。30~70行产生8个页面，每个页面中的图形相同，但在屏幕上的显示位置不同。

90~140行顺次显示这8个页面。由90行语句控制，反复显示10次，最后由150行语句将活动页和可见页都置0，恢复原值。

2. 由 GET 和 PUT 语句实现快速动画

图形 GET 语句用于从屏幕上指定的某一矩形区域中，读出一些点，并把它存入数组中；而图形 PUT 语句则把以前存入数组的点写到屏幕上指定的矩形区域中。因而，利用 GET 语句和 PUT 语句可以实现快速动画。

GET 语句的格式：

GET (X1, Y1) - (X2, Y2), 数组名

参数 (X1, Y1)、(X2, Y2) 是指定矩形区域两个对顶角点的坐标。可以用绝对坐标，也可以用相对坐标。

数组名是一个一维数值数组的名称，数组的大小应能容纳矩形区域内所有象元的彩色码。数组的最小容量（以字节为单位）可用下式计算

$$4 + \text{INT}((H * \text{BITS} + 7) / 8) * W$$

式中，H 和 W 分别是矩形的高度和宽度，BITS 是每个象

元所对应的二进制数的位数。在中分辨率时，BITS=2，高分辨时 BITS=1。附加的 4 个字节用于存放 H 和 W 的数值，其中两个字节存 H，另两个字节存放 W。

例如：

GET (50, 50) - (69, 79), A

其中矩形宽 20 点，高 30 点，数组 A 需要字节数为

$$4 + \text{INT}((20 \times 2 + 7) / 8) \times 30 = 154$$

PUT 语句格式：

PUT (X, Y), 数组名, 操作

(X, Y)为矩形区域左上角的坐标。数组名和 GET 语句中的数组名相同。

操作有五种，即 PSET, PRESET, AND, OR, XOR。缺省时自动执行 XOR 操作。各个操作的意义如下：

PSET 出现在屏幕上的图象和存放在数组中的图象相同。

PRESET 屏幕上的图象和存在数组中的图象相反。即屏幕上的黑点对应于存放在数组中的白点；而屏幕上的白点对应于存放在数组中的黑点。

AND 只有当屏幕上的点已经是白的、而且数组中的点也是白的，屏幕上的点才是白的。

OR 如果屏幕上的点是白的，或者数组中的对应点是白的，或者两者都是白的、则屏幕上的点是白的。

XOR 存放在数组中的每一个黑点，都会使屏幕上的对应点保持不变，而存放在图象中的每一个白点；都会使屏幕上的对应点改变颜色，即白的变为黑的、黑的变为白的。

因而对 AND、OR、XOR 来说，数组中的图象和已经在屏幕上的图象，都会影响屏幕上的图象。

在中分辨率图形方式下，PRESET、AND、OR、XOR 操作是比较复杂的，因为它包括了四种颜色，而不仅仅是黑和白两种颜色。在 BASIC 手册中给出了在中分辨率情况下，AND OR、和 XOR 对四种颜色的作用的表格。为了获得每种操作的视觉效果，可能需要进行大量的实验。

XOR 操作对于动画是非常有用的，因为同样的语句既能写入图象，也能擦掉图象。两个带有 XOR 的 PUT 语句，从同一数组写入屏幕的同一区域时，则相互抵消，而屏幕上保持不变。例如，执行下列两个语句：

```
PUT (100, 50), A, XOR
```

```
PUT (100, 50), A, XOR
```

则执行完第二个 PUT 语句后，和执行第一个 PUT 语句以前屏幕上显示的结果是一样的，不论存放在数组 A 中的图象或者占据屏幕上对应区域的图象有多么复杂，这都是对的。但是，利用 XOR 时，存放在 A 中的图象不能取代屏幕上对应区域的图象，而仅仅是修改颜色，因此在背景上还留下某些使用 XOR 的痕迹，这就使 XOR 在某些场合下的应用受到限制。

例 1-37 彩球弹跳

```
1000 'pcl-37
```

```
1420 DEFINT A-Z: DIM B(280): DIM A(150)
```

```
1430 CLS:SCREEN 1, 0: COLOR 8, 0: KEY OFF
```

```
1440 CIRCLE (160, 100), 10, 2
```

```
1450 PAINT (160, 100), 2, 2
```

```
1460 GET (150, 90)-(170, 110), A
```



```

1470 CLS: LINE (19, 0)-(299, 177), , B
1480 LINE (20, 1)-(300, 178), , B
1490 LOCATE 24, 11: PRINT " Press ESC to exit ";
1500 FOR X=20 TO 280 STEP 4
1510 B(X)=159-CINT(ABS(SIN(X * 7.853981E-02) * X)\2)
1520 NEXT : Y2=150
1530 LINE (223, 3)-(253, 175), 3, BF
1540 LINE (160, 3)-(190, 175), 2, BF
1550 LINE (100, 3)-(130, 175), 1, BF
1560 FOR D=0 TO 1
1570 S=20+D * 260: E=280-D * 260
1580 FOR X=S TO E STEP 4-8 * D
1585 FOR I=1 TO 50: NEXT I
1590 Y=B(X): N+(170-Y2)\5: IF F THEN PUT (X2, Y2),
    A
1600 PUT (X, Y), A: X2=X: Y2=Y: F=-1: NEXT
1610 NEXT
1615 A$=INKEY$: IF A$=CHR$(27) THEN CLS:
    END
1620 IF A$ < > " " THEN 1615, ELSE GOTO 1560

```

程序运行后，从屏幕上会看到一个彩球在三条彩带之间弹跳的美丽画面。

例 1-38 太空船

```

1300 'pc1-38
1400 KEY OFF
1410 DEFINT I-N: DIM I (800)
1420 SCREEN 1, 0: COLOR 8,0

```

```

1430 CLS: CIRCLE (160,100), 30, 1, , , 45: PAINT(160, 100),
      1,1: DRAW"bm160, 100c30bm160, 10 oh30": LINE (130,
      100)-(190,100),2: GET(130,70)-(190, 130), I
1440 LINE (0,0)-(100,199), 0, BF: LINE (101, 0)-(200, 199), 2,
      BF: LINE (201, 0)-(300, 199), 3, BF
1450 LOCATE 23, 2: PRINT "Press ESC";
1460 LOCATE 24, 2: PRINT "to exit";
1470 FOR J=2 TO 6
1480 K1=RND*259: K2=RND*138: PUT(K1, K2), I, XOR:
      FOR I1=1 TO 150: NEXT: PUT (K1, K2), I, XOR: NEXT
1490 A$ =INKEY$: IF A$ =CHR$(27) THEN CLS:
      END
1500 IF A$ < > " THEN 1490 ELSE 1470

```

1430 行语句画太空船，并将其存入数组 I。1440 行语句将屏幕划分三块矩形彩带。1450 行~1460 行语句显示字符“press ESC to exit”。1470~1480 行语句用两个 PUT (K1, K2), I, XOR 语句来显示和擦掉屏幕中任一位置上的太空船。K1, K2 为屏幕显示范围内的随机数坐标。程序执行后画面如图 1.12。



第二部分 平面图形程序

第二章 图案变换及操作

在计算机辅助图案设计时常常要对屏幕显示图形进行各种操作。例如，在屏幕上移动某些图形的位置、修改图形尺寸、变化图案形状、将屏幕显示图案存入磁盘及用绘图机输出图形等。下面，我们分别讨论各种图形操作方法。

2.1 平移变换

图形平移实际上是一种坐标变换。如图 2.1 所示，当图形上点 (X, Y) 沿 X 轴平移 T_x 、沿 y 轴平移 T_y ，移动后的点坐标为 (X', Y') 可表示为：

$$X' = X + T_x$$

$$Y' = Y + T_y$$

对一个图形或图形的一部分实行平移时，在程序中必须完成如下几点：

- (1) 计算出被平移图形中每一点的新坐标 (X', Y') ；
- (2) 擦去原来的图形；
- (3) 按新的坐标位置画出图形。

由于图形常常是由一些点、线或圆弧构成的，因而平移并不需要对每一个点都进行计算，只要对线段的端点、矩形的顶点、或圆弧中心及一些孤立点进行新坐标计算，然后利

用 PSET、LINE、CIRCLE 等图形语句在新的位置上画出图形。

平移可以在世界坐标系内进行，也可以在点坐标系内进行。

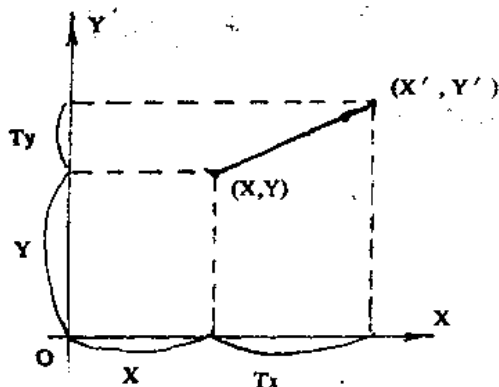


图 2.1

例 2-1 画一个三角形并将三角形进行平移的程序

```

5  'pc2-1
10  READ PX1, PX2, PX3, PY1, PY2, PY3
20  GOSUB 100
30  FOR J=0 TO 100: NEXT J
40  TX=50: TY=30
50  GOSUB 200
60  GOSUB 100
70  END

100
110 SCREEN 1, 0: CLS
120 LINE (PX1, PY1)-(PX2, PY2), 1

```

```

130 LINE-(PX3, PY3), 1
140 LINE-(PX1, PY1), 1
150 RETURN
200 '
210 PX1=PX1+TX: PY1=PY1+TY
220 PX2=PX2+TX: PY2=PY2+TY
230 PX3=PX3+TX: PY3=PY3+TY
240 RETURN
300 DATA 100, 220, 160, 100, 100, 50

```

说明:

这个程序中给出的数据，是点坐标系的数据，平移在点坐标系内进行。

10 行 读入数据;
 20 行 调用 100 行的子程序画三角形;
 40 行 给出 Tx, Ty 的数值;
 50 行 调用 200 行的子程序进行新坐标计算;
 60 行 再次调用 100 行的子程序按新坐标画出三角形。

如果去掉 110 行的语句 CLS，则在屏幕上显示出平移前的三角形和平移后的三角形。

例 2-2 以数学公式计算四瓣花型单元数据，在正方形内配置为 6×6 个单元的花型图案的程序。

程序中花型单元的平移，是由两个 FOR~NEXT 语句实现的。

```

100 'pc2-2
110 SCREEN 2: CLS
120 PI=3.14159: R=30

```

```

130 FOR PX=170 TO 470 STEP 60
140 FOR PY=50 TO 350 STEP 60
150   GOSUB 200
160 NEXT PY
170 NEXT PX
180 END
200 FOR A=0 TO 2*PI STEP PI/16
210 D=R*(1+1/5*SIN(12*A))
220 T=D*(1/2+1/2*SIN(4*A))
230 X1=T*COS(A): Y1=-T*SIN(A)
240 X=(PX+X1)/2: Y=(PY+Y1)/2
250 IF A=0 THEN PSET(X,Y): BX=X: BY=Y
260 LINE-(X,Y)
270 NEXT A: LINE-(BX,BY)
280 RETURN

```

程序运行结果如图 2.2.

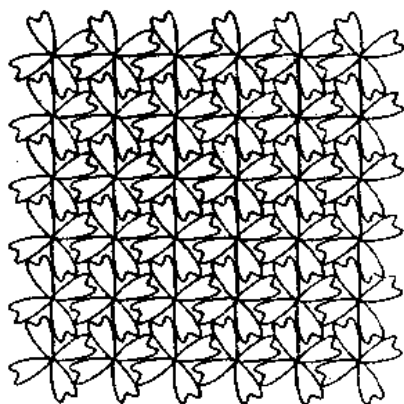


图 2.2

2.2 比例变换

以坐标原点 O 为基准，将物体沿 X 轴放大 / 缩小 S_x 倍、沿 Y 轴方向放大 / 缩小 S_y 倍，这时图形上的点 $P(X, Y)$ 移到 $P'(X', Y')$ ，即

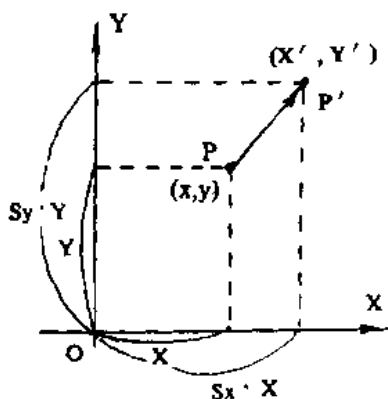


图 2.3

$$X' = X \cdot S_x$$

$$Y' = Y \cdot S_y$$

如图 2.3 所示。

当 $S_x = S_y$ 时，原点 O 、 P 、 P' 排在一条直线上，图形只改变大小，不变化形状；当 $S_x \neq S_y$ 时，则点 O 、 P 、 P' 不在一条直线上，这时图形不仅变化大小，而且变化形状。注意的是，这里比例变换图形的中心点是坐标原点。实际上，相对于屏幕上的某一点也可以进行图形的比例变换。例如，相对于参考点坐标 (X_0, Y_0) 的比例变换公式为：

$$X' = X \cdot S_x + X_0(1-S_x)$$

$$Y' = Y \cdot S_y + Y_0(1-S_y)$$

与平移一样，对一个图形进行比例变换时，只要对线段的始点、终点坐标进行计算就行了。

例 2-3 对三角形进行比例变换的程序

```

5  'pc2-3
10 READ PX1,PX2,PX3,PY1,PY2,PY3
20 GOSUB 100
30 FOR J=0 TO 100:NEXT J
40 SX=.5:SY=.5:X0=160:Y0=100
50 GOSUB 200
60 GOSUB 100
70 END
100 '
110 SCREEN 1,0:CLS
120 LINE (PX1,PY1)-(PX2,PY2),1
130 LINE-(PX3,PY3),1
140 LINE-(PX1,PY1),1
150 RETURN
200 '
210 PX1=PX1 * SX+X0 * (1-SX) : PY1=PY1 * SY
    +Y0 * (1-SY)
220 PX2=PX2 * SX+X0 * (1-SX) : PY2=PY2 * SY
    +Y0 * (1-SY)
230 PX3=PX3 * SX+X0 * (1-SX) : PY3=PY3 * SY
    +Y0 * (1-SY)
240 RETURN

```


300 DATA 100, 220, 160, 100, 100, 50

程序中变换的参考点为 (160, 100)，试运行这个程序，观察屏幕上显示的图形。

去掉 110 行中的 CLS 语句，并变化 40 行语句中的 S_x 、 S_y 值，运行程序，观察

(1) $S_x > 1$, $S_y > 1$;

(2) $S_x < 1$, $S_y < 1$;

(3) $S_x = S_y$

(4) $S_x \neq S_y$

等各种不同情况下图形的大小和形状的变化。

2.3 旋转变换

图形旋转是将屏幕上的图形绕某一点旋转一个角度。例如，如图 2.4 所示，将点 (X, Y) 以坐标原点为中心，顺

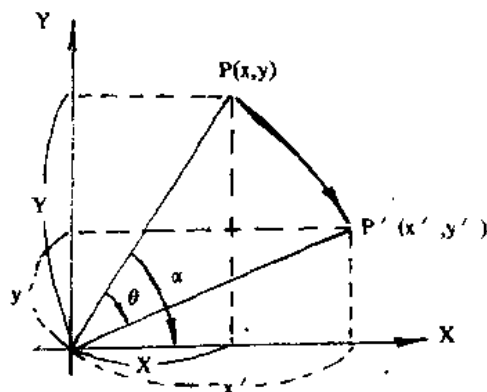


图 2.4

时针旋转 θ 时，则点 P (X, Y) 移到 P'(X', Y')。设点

P (X, Y) 与原点 O 的连线 OP 与 X 轴夹角为 α , 则点 P (X, Y) 可表示为

$$x = \sqrt{x^2 + y^2} \cos \alpha = \gamma \cdot \cos \alpha$$

$$y = \sqrt{x^2 + y^2} \sin \alpha = \gamma \cdot \sin \alpha$$

$$\gamma = \sqrt{x^2 + y^2}$$

旋转 θ 角度后点 P' (x' , y') 的坐标

$$x' = \gamma \cdot \cos(\alpha - \theta)$$

$$y' = \gamma \cdot \sin(\alpha - \theta)$$

将上式写成三角函数的展开式:

$$x' = \gamma(\cos \alpha \cdot \cos \theta + \sin \alpha \cdot \sin \theta)$$

$$y' = \gamma(\sin \alpha \cdot \cos \theta - \cos \alpha \cdot \sin \theta)$$

将 $x = \gamma \cdot \cos \alpha$, $y = \gamma \cdot \sin \alpha$

$$x' = x \cdot \cos \theta + y \cdot \sin \theta$$

$$y' = -x \cdot \sin \theta + y \cdot \cos \theta$$

这样, 旋转后点坐标 P' (X' , Y') 只与旋转前的坐标 (X, Y) 和旋转角度 θ 有关, 这在编写程序时就方便多了。

如果旋转变换的基准点不是坐标原点而选择为 (X_0 , Y_0) 时, 当旋转角度为 A 时, 则:

$$XR = X_0 + (X - X_0)\cos(A) + (Y - Y_0) \cdot \sin(A)$$

$$YR = Y_0 + (Y - Y_0)\cos(A) - (X - X_0) \cdot \sin(A)$$

这时, 其准点 (X_0 , Y_0) 可以选择屏幕内一点, 也可以选择屏幕外一点。

例 2-4 将三角形进行旋转的程序

```
5 'pc2-4
```

```
10 READ PX1,PX2,PX3,PY1,PY2,PY3
```

```

20 GOSUB 100
30 FOR J=0 TO 100 :NEXT J
40 A=.3
50 GOSUB 200
60 GOSUB 100
70 END
100 '
110 SCREEN 1,0 :CLS
120 LINE(PX1,PY1)-(PX2,PY2), 1
130 LINE-(PX3, PY3),1
140 LINE-(PX1, PY1)-(PX2, PY2), 1
150 LINE-(PX3, PY3), 1
160 LINE-(PX1, PY1), 1
170 RETURN
200 '
210 PX1=PX1 * COS(A)+PY1 * SIN(A)
220 PY1=PY1 * COS(A)-PX1 * SIN(A)
230 PX2=PX2 * COS(A)+PY2 * SIN(A)
240 PY2=PY2 * COS(A)-PX2 * SIN(A)
250 PX3=PX3 * COS(A)+PY3 * SIN(A)
260 PY3=PY3 * COS(A)-PX3 * SIN(A)
270 RETURN
300 DATA 100, 220, 160, 100, 100, 50

```

这个程序的旋转基准点为坐标原点。当旋转基准点选择为屏幕中心（160，100）时，其程序如下：

例 2-5 将三角形以屏幕中心为旋转基准点的程序

```

5 'pc2-5

```

```

10 READ PX1,PX2,PX3, PY1, PY2, PY3
20 GOSUB 100
30 FOR J=0 TO 100:NEXT J
40 A=.3:X0=160:Y0=100
50 GOSUB 200
60 GOSUB 100
70 END
100 /
110 SCREEN 1,0:CLS
120 LINE (PX1, PY1)-(PX2, PY2), 1
130 LINE-(PX3, PY3),1
140 LINE-(PX1, PY1), 1
150 RETURN
200 /
210 PX1=X0+(PX1-X0)*COS(A)+(PY1-Y0)*SIN(A)
220 PY1=Y0+(PY1-Y0)*COS(A)-(PX1-X0)*SIN(A)
230 PX2=X0+(PX2-X0)*COS(A)+(PY2-Y0)*SIN(A)
240 PY2=Y0+(PY2-Y0)*COS(A)-(PX2-X0)*SIN(A)
250 PX3=X0+(PX3-X0)*COS(A)+(PY3-Y0)*SIN(A)
260 PY3=Y0+(PY3-Y0)*COS(A)-(PX3-X0)*SIN(A)
270 RETURN
300 DATA 100, 220, 160, 100, 100, 50

```

例 2-6 旋转小正方形程序

```

100 'pc2-6
110 SCREEN 2:CLS
120 PI=3.14159:T=SQR(170^2*2)
130 FOR PX=-170 TO 170 STEP 20

```

```

140 FOR PY=-170 TO 170 STEP 20
150 A=SQR(PX^2+PY^2)/T*PI/2+PI/4
160 FOR AL=0 TO 2*PI STEP PI/2
170 X1=10*COS(AL):Y1=10*SIN(AL)
180 GOSUB 250
190 X=PX+320+X2:Y=PY+200-Y2
200 IF AL=0 THEN PSET(X,Y)
210 LINE-(X,Y)
220 NEXT AL,PY,PX
230 END
250 X2=X1*COS(A)-Y1*SIN(A)
260 Y2=X1*SIN(A)+Y1*COS(A)
270 RETURN

```

程序运行结果如图 2.5。150 行语句中的 A 式决定各个小正方形的旋转角度。A 与图形中心到旋转正方形中心的距离有关。变化旋转角度 A 的表达式。例如，当 A 取

$$A = \text{RND} * P$$

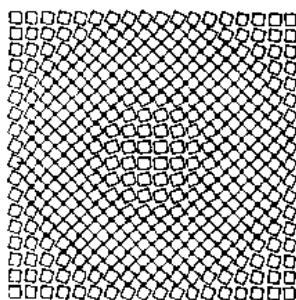


图 2.5

时，程序为：

```

100 'pc2-7
110 SCREEN 2: CLS
120 PI=3.14159: T=SQR(170^2 * 2)
130 FOR PX=-170 TO 170 STEP 20
140 FOR PY=-170 TO 170 STEP 20
150 A=RND * PI
160 FOR AL=0 TO 2 * PI STEP PI / 2
170 X1=10 * COS(AL): Y1=10 * SIN(AL)
180 GOSUB 250
190 X=PX+320+X2:Y=PY+200-Y2
200 IF AL=0 THEN PSET (X, Y)
210 LINE-(X, Y)
220 NEXT AL, PY, PX
230 END
250 X2=X1 * COS(A)-Y1 * SIN(A)
260 Y2=X1 * SIN(A)+Y1 * COS(A)
270 RETURN

```

程序运行结果如图 2.6。此程序中删去了 120 行语句中的 T 表达式。

若将 150 行改为:

```
150 A=ABS(PX)/170 * PI / 2
```

则程序为:

```

100 'pc2-8
110 SCREEN 2: CLS
120 PI=3.14159: T=SQR(170^2 * 2)
130 FOR PX=-170 TO 170 STEP 20
135 A=ABS(PX)/170 * PI / 2

```

```

140 FOR PY=-170 TO 170 STEP 20
160 FOR AL=0 TO 2*PI STEP PI/2
170 X1=10*COS(AL):Y1=10*SIN(AL)
180 GOSUB 250
190 X=PX+320+X2:Y=PY+200-Y2
200 IF AL=0 THEN PSET (X, Y)
210 LINE-(X, Y)
220 NEXT AL, PY, PX
230 END
250 X2=X1*COS(A)-Y1*SIN(A)
260 Y2=X1*SIN(A)+Y1*COS(A)
270 RETURN

```

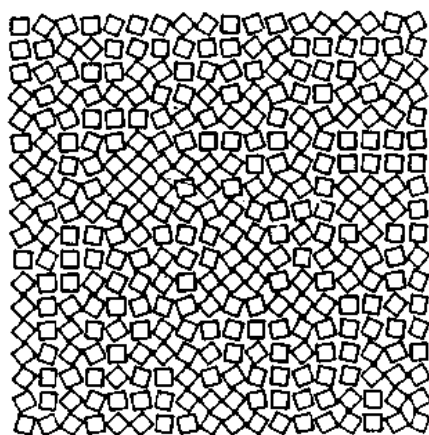


图 2.6

运行结果如图 2.7 所示。

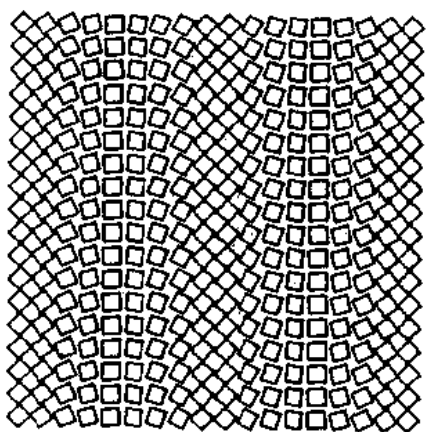


图 2.7

2.4 对称变换和错切变换

在前述图形变换中，我们没有采用变换矩阵，实际上，利用矩阵描述图形变换是很方便的。例如，利用矩阵

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

将点坐标(X, Y)和矩阵相乘，则可得新点的坐标：

$$[XY] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [aX + cY \quad bX + dY]$$

矩阵元素 a、b、c、d 取不同的值，可以实现不同的变换。

采用矩阵参数的图形变换试验程序为：

```
100 'pc2-9
```

```
110 SCREEN 2:CLS
```



```

120 FOR X=20 TO 620 STEP 100
130 LINE (X,0)-STEP (0, 199), , , &H8888: NEXT
140 FOR Y=0 TO 199 STEP 50
150 LINE (0,Y)-STEP (639,0), , , &H8888: NEXT
160 LINE (320, 0)-(320,199): LINE(0, 100)-(639,100)
210 INPUT "a, b, c, d <= 2"; A, B, C, D: CLS
220 PRINT A: B
230 PRINT C: D
240 READ X, Y
250 IF X=999 THEN GOTO 300
260 IF X=111 THEN READ X, Y: GOSUB 310: PSET (XX
    YY): GOTO 240
270 GOSUB 310
280 LINE-(XX, YY)
290 GOTO 240
300 END
310 XX=A * X+C * Y+320: YY=100-(B * X+D * Y) / 2
320 RETURN
330
340 DATA 111, 111, 10, 1, 40, 1, 40, 40, 50, 40, 25
350 DATA 50, 1, 40, 10, 40, 10, 1, 111, 111, 35, 20
360 DATA 111, 111, 35, 20, 65, 20, 65, 60, 75, 60
370 DATA 50, 70, 25, 60, 35, 60, 35, 20, 111, 111
380 DATA 60, 40, 90, 40, 90, 80, 100, 80, 75, 90
390 DATA 50, 80, 60, 80, 60, 40, 999, 999

```

这是一个画箭矢图的程序。由 210 行语句从键盘输入 a、b、c、d 各矩阵元素的数值，从而实现不同的图形变换。

一、画两个箭矢

取矩阵参数为:

A	B	C	D
1	0	0	1

时, 运行程序, 屏幕显示图形如图 2.8。称为原型图形。

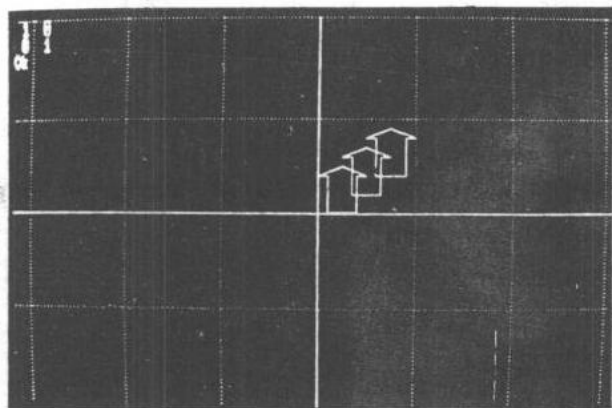


图 2.8

二、放大

取矩阵参数为:

A	B	C	D
1	0	0	2

时, 图形纵向扩大 2 倍。如图 2.9 所示。

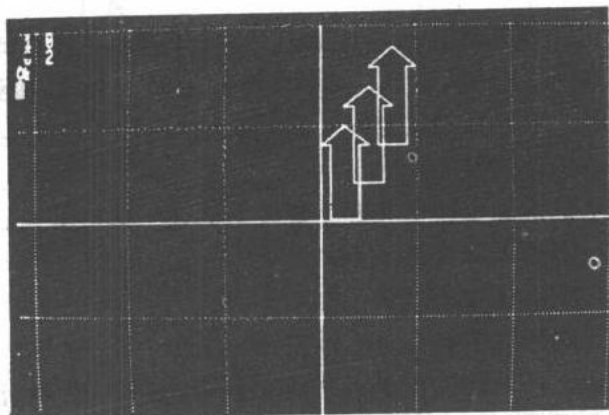


图 2.9

若取矩阵参数值为:

A	B	C	D
2	0	0	2

时, 程序运行结果如图 2.10 所示, 图形在横向扩大 2 倍.

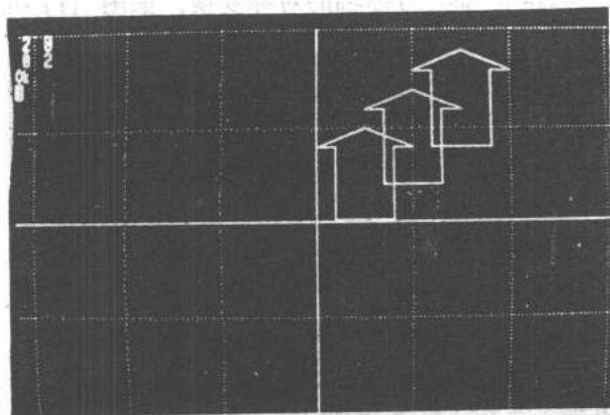


图 2.10

不难得出，矩阵参数 A 、 D 为图形放大参数， A 、 D 均大于 1 时，图形被放大， A 、 D 均小于 1 时，图形被缩小。 $A = D$ 时，图形做相似变换， $A \neq D$ 时，图形形状变化。

三、对称变换

图形的对称变换可在四个象限内实现，即可实现。

- (1) 图形以 X 轴为对称轴；
- (2) 图形以 Y 轴为对称轴；
- (3) 图形以原点为对称轴。

实际上，对称变换是通过将变换矩阵中 A 或 D 元素值取反来实现的。

例如，当在实验程序中输入参数为

A	B	C	D
-2	0	0	2

时，可实现以 Y 轴为对称轴的对称变换。如图 2.11 所示。

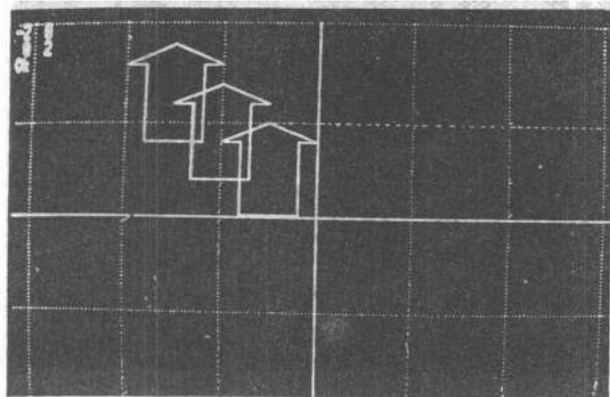


图 2.11

若输入

A	B	C	D
2	0	0	-2

则如图 2.12, 实现以 X 轴为对称的变换。

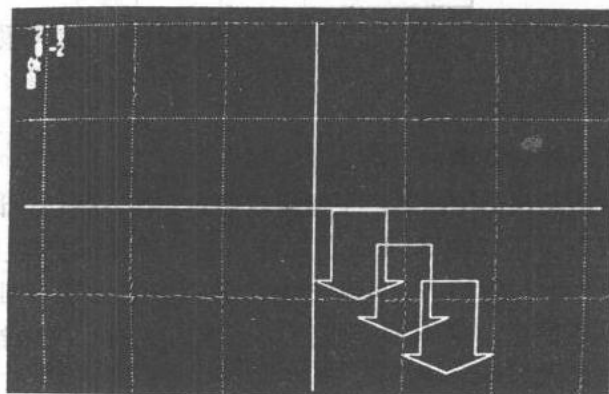


图 2.12

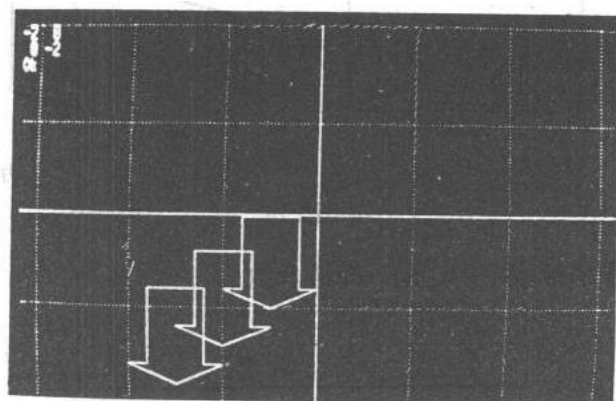


图 2.13

若输入

A	B	C	D
-2	0	0	-2

时，运行程序，如图 2.13 所示，实现以原点为对称的变换。

四、错切变换

错切变换是使图形产生错切变形的变换。错切一词来源于结构力学。如图 2.14 所示那样，在一块矩形材料上，施加两个方向相反的力，矩形材料产生变形变成平行四边形。在图形变换中，使图形产生错切变形的变换，叫做错切变换。

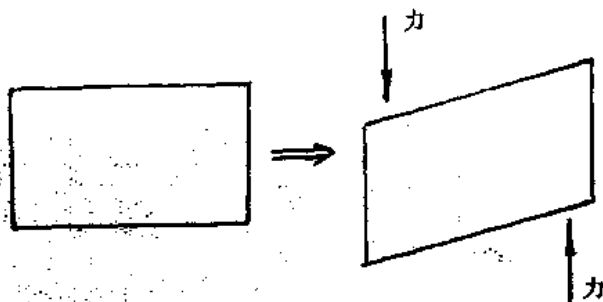


图 2.14

在试验程序中，输入矩阵参数为

A	B	C	D
1	0.5	0	1

时，则

$$\begin{bmatrix} X & Y \end{bmatrix} \cdot \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} X & 0.5X + Y \end{bmatrix}$$

X 的值不变, 新坐标 Y' 的值, 为 X 乘上 B 再加上 Y 。这时, 显示图形如图 2.15。

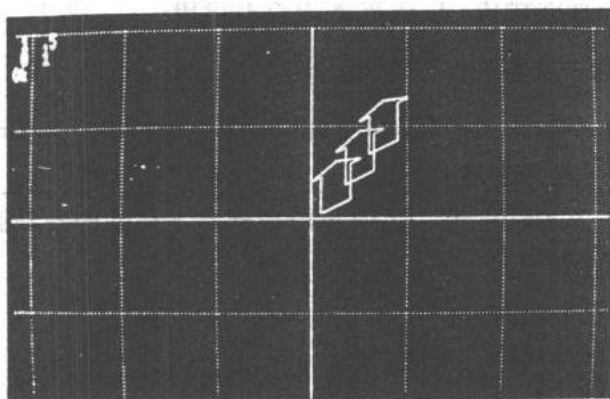


图 2.15

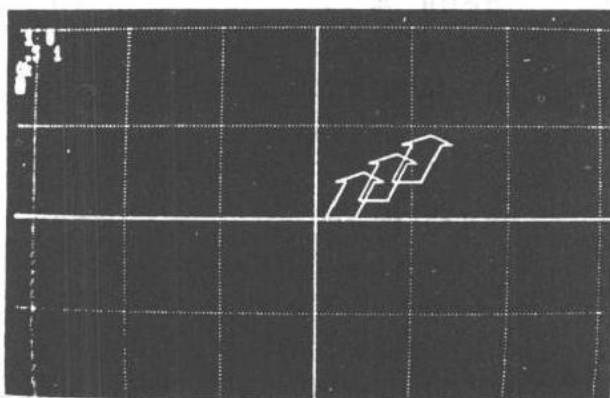


图 2.16

若输入参数为:

A	B	C	D
1	0	0.5	1

时, 屏幕图形如图 2.16.

从图中看出, B、C 元素为错切变换的关键元素, B 产生 Y 方向的错切, C 决定 X 方各的错切.

2.5 渐变图案

描述由一个图形变化为另一个图形过程中的各个中间图形, 称为渐变图形.

例 2-10: 画由左上抛物线到右下角圆的各渐变中间图形程序

```
100 pc2-10
110 SCREEN 2 :CLS
120 DIM X1(120), X2(120), Y1(120), Y2(120)
130 PI=3.14159:N=-1:NN=35
140 GOSUB 520
150 GOSUB 560
160 FOR PX=170 TO 470 STEP 60
170 FOR PY=50 TO 350 STEP 60
180 N=N+1
190 FOR I=1 TO 120
200 X=(X2(I)-X1(I))/NN*N+X1(I)
210 Y=(Y2(I)-Y1(I))/NN*N+Y1(I)
220 X=(X+PX):Y=(Y+PY)
230 IF I=1 THEN PSET (X, Y/2):BX=X:BY=Y/2
```



```

240     LINE-(X, Y / 2)
250     NEXT I:LINE-(BX, BY)
260 NEXT PY, PX
280 END

520     FOR A=0 TO 2 * PI / 60:I = I+1
530     X1(I) = 28 * COS(a):Y1(I) = 14 * COS(2 * A)
540     NEXT A:I=0:RETURN
560     FOR A=0 TO 2 * PI STEP PI / 60:I = I+1
570     X2(I) = 28 * COS(A):Y2(I) = 14 * SIN(A)
580     NEXT A:RETURN

```

程序运行结果如图 2.17.

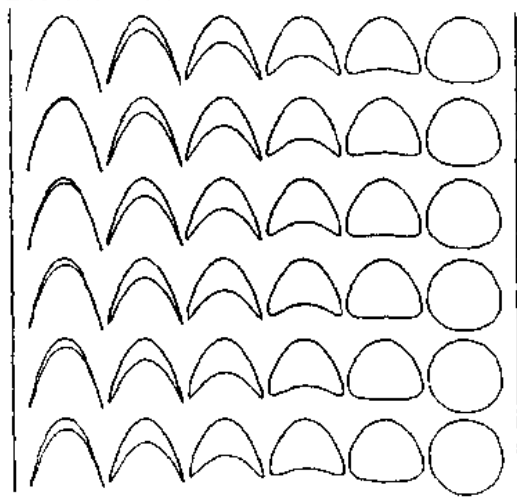


图 2.17

渐变图形是两个图形都取相同的公式化图形数据数绘出的。左上角的抛物线，实际上是在一条曲线上往复画出的闭合曲线。抛物线和圆均在 $0 \sim 2\pi$ 范围内取相同的步长角和

点数据数。

140~150 行, 转入子程序计算图形的点坐标值;

160~170 行, 计算各中间图形的配置位置。

190~250 行, 计算各中间图形的坐标值。

例 2-11: 由蝶型变化为首蓓叶形。

```
100 pc2-11
110 SCREEN 2 :CLS
120 DIM X1(120), X2(120), Y1(120), Y2(120)
130 PI=3.14159:N=-1:NN=35
140 GOSUB 510
150 GOSUB 560
160 FOR PX=170 TO 470 STEP 60
170 FOR PY=50 TO 350 STEP 60
180 N=N+1
190 FOR I=1 TO 120
200 X=(X2(I)-X1(I))/NN*N+X1(I)
210 Y=(Y2(I)-Y1(I))/NN*N+Y1(I)
220 X=(X+PX)/2:Y=(Y+PY)/2
230 IF I=1 THEN PSET (X, Y):BX=X:BY=Y
240 LINE-(X, Y)
250 NEXT I:LINE-(BX, BY)
260 NEXT PY, PX
270 LINE(130, 10)-(510, 190), , B
280 END
510 R=35
520 FOR A=0 TO 2*PI STEP PI/60:I=I+1
530 X1(I)=1.1*(R/5*SIN(8*A)+R*SIN(2*A))*COS(A)
```

```

535  Y(I)=0.85*(R/5*SIN(8*A)+R*SIN(2*A))*SIN(A)
540  NEXT A:I=0:RETURN
560  FOR A=0 TO 2*PI STEP PI/60:I=I+1
570  X2(I)=(R/5*SIN(8*A)+R*SIN(2*A))*COS(A)
575  Y2(I)=(R/5*SIN(8*A)+R*SIN(2*A))*SIN(A)
580  NEXT A:RETURN

```

程序运行结果如图 2.18.

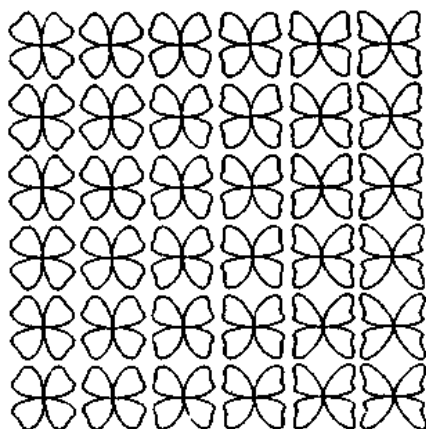


图 2.18

例 2-12 由圆型渐变为正方形

```

100 'pc2-12
110 SCREEN 2 :CLS
120 DIM X1(120), X2(120),Y1(120),Y2(120)
130 PI=3.14159:NN=20
140 GOSUB 520:GOSUB 580
150 FOR N=0 TO NN
160 PX=60*N/NN+320

```

```

170     FOR I=1 TO 120
180     X=(X2(I)-X1(I))/NN*N+X1(I)
190     Y=(Y2(I)-Y1(I))/NN*N+Y1(I)
200     X=X+PX:Y=Y/2+100
210     I=1 THEN PSET (X, Y):BX=X:BY=Y
220     LINE-(X, Y)
240     NEXT I:LINE-(BX, BY)
250 NEXT N
260 END
520 R=180
530 FOR A=0 TO 2*PI STEP PI/60
540     I=I+1
550     X1(I)=R*COS(A):Y1(I)=-R*SIN(A)
560 NEXT A:A=0:RETURN
580 M=4:K=30:R=90:L=2*R*SIN(PI/M):DD=L/K
590 FOR A=0 TO 2*PI-2*PI/M STEP 2*PI/M
600     AA=A+PI/2
610     X0=R*SIN(AA):Y0=R*COS(AA)
620     FOR I=0 TO K-1 :J=J+1
630         X2(J)=X0+I*DD*SIN(AA+PI/2+PI/M)
640         Y2(J)=Y0+I*DD*COS(AA+PI/2+PI/M)
650     NEXT I:A:RETURN

```

程序运行结果如图 2.19

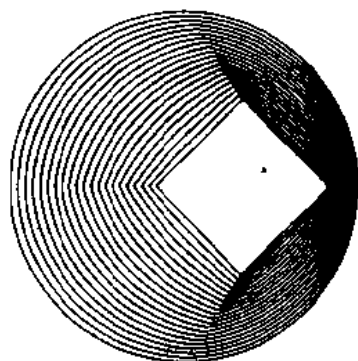


图 2.19

例 2-13 由苜蓿叶渐变到圆

```

100 pc2-13
110 SCREEN 2 CLS
120 DIM X1(120), X2(120), Y1(120), Y2(120)
130 PI=3.14159:NN=25
140          GOSUB 520:GOSUB 570
150 FOR N=0 TO NN
160     FOR I=1 TO 120
170         X=(X2(I)-X1(I))/NN*N+X1(I)
180         Y=(Y2(I)-Y1(I))/NN*N+Y1(I)
190         X=X+320:Y=Y+100
200         IF I=1 THEN PSET (X, Y):BX=X:BY=Y
210         LINE-(X, Y)
220     NEXT I:LINE-(BX, BY)
230 NEXT N

```

```

240 END
520 R=180
530 FOR A=0 TO 2*PI STEP PI/60:I=I+1
540 X1(I)=R*COS(A):Y1(I)=-R*SIN(A)/2
550 NEXT A:I=0:RETURN
570 R=100
580 FOR A=0 TO 2*PI STEP PI/60:I=I+1
590 D=R*(1+1/2*SIN(12*A))
600 T=D*(1/2+1/2*SIN(4*A))
610 X2(I)=T*COS(A):Y2(I)=-T*SIN(A)/2
620 NEXT A:RETURN

```

程序运行结果如图 2.20

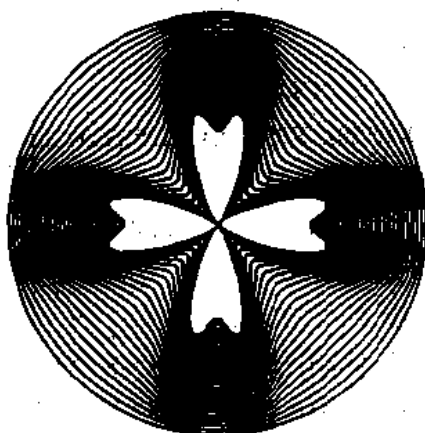


图 2.20

例 2-14 渐变图形间隔按对数规律分布

```

100 pc2-14
110 SCREEN 2 :CLS

```

```

120 DIM X1(120), X2(120), Y1(120), Y2(120)
130 PI=3.14159:NN=25
140 GOSUB 520:GOSUB 580
150 FOR N=0 TO NN
160 FOR I=1 TO NN
170 X=(X2(I)-X1(I))/LOG(NN+1)*LOG(N+1)+X1(I)
180 Y=(Y2(I)-Y1(I))/LOG(NN+1)*LOG(N+1)+Y1(I)
190 X=X+320:Y=Y+200
200 IF I=1 THEN PSET (X, Y/2):BX=X:BY=Y/2
210 LINE-(X, Y/2)
220 NEXT I:LINE-(BX, BY)
230 NEXT N
240 END
520 R=50
530 FOR A=0 TO 2*PI STEP PI/60:I=I+1
540 D=R*(1+SIN(5*(A+PI/4)))
550 X1(I)=D*COS(A+PI/4):Y1(I)=-D*SIN(A+PI/4)
560 NEXT A:I=0:RETURN
580 R=250:M=4:K=30:L=2*R*SIN(PI/M):DD=L/K
590 FOR A=0 TO 2*PI-2*PI/M STEP 2*PI/M
600 AA=A+3*PI/4
610 X0=R*SIN(AA):Y0=R*COS(AA)
620 FOR I=0 TO K-1:J=J+1
630 X2(J)=X0+I*DD*SIN(AA+PI/2+PI/M)
640 Y2(J)=Y0+I*DD*COS(AA+PI/2+PI/M)
650 NEXT I,A:RETURN

```

程序运行结果如图 2.21。

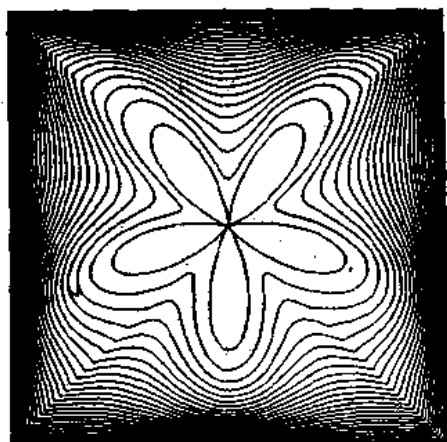


图 2.21

2.6 扇形变换

将图 2.22 所示的左侧长方形的图形变化为右侧那样的扇形图形的变换叫做扇形变换。原型图中任意点的 $P(X_1, Y_1)$ 经过扇形变换后，点 $P_2(X_2, Y_2)$ 的坐标为：

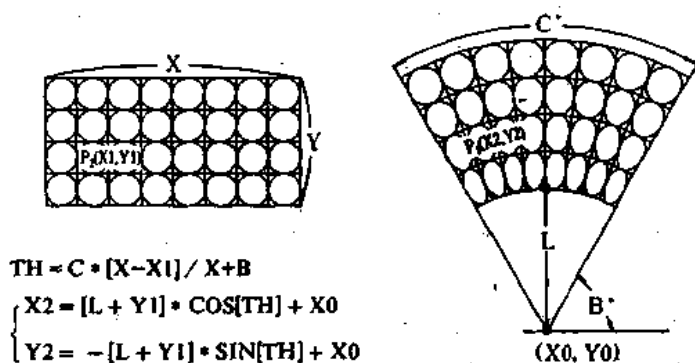


图 2.22

$$X_2 = (L + Y_1) * \cos(\text{TH}) + X_0$$

$$Y_2 = -(L + Y_1) * \sin(\text{TH}) + Y_0$$

式中,

$$\text{TH} = C * (X - X_1) / X + B$$

X_0, Y_0 为扇形圆心坐标, L 为扇形半径。

例 2-15 将 4×8 个纵横排列的 4 瓣花型矩形图案变换为扇形图案

```

100 'pc2-15
110 SCREEN 2 :CLS
120 PI=3.14159:X=640:Y=300:L=280
130 X0=320:Y0=600:C=PI/3:R=X/9
140 FOR PX=X/16 TO 15*X/16+X/16 STEP X/8
150     FOR PY=Y/8 TO 7*Y/8 STEP Y/4
170         FOR A=0 TO 2*PI+PI/60 STEP PI/64
180             D=R*(4/5+1/5*SIN(12*A))
190             T=D*(0.5+0.5*SIN(4*A))
200             X1=T*COS(A)+PX
210             Y1=T*SIN(A)+PY
220             GOSUB 280
230             IF A=0 THEN PSET(X2,Y2/2)
240             LINE-(X2,Y2/2)
250         NEXT A, PY, PX
260 END

```

```

280  TH = C * (X - X1) / X + B
290  X2 = (L + Y1) * COS(TH) + X0
300  Y2 = -(L + Y1) * SIN(TH) + Y0
310  RETURN

```

程序运行结果如图 2.23

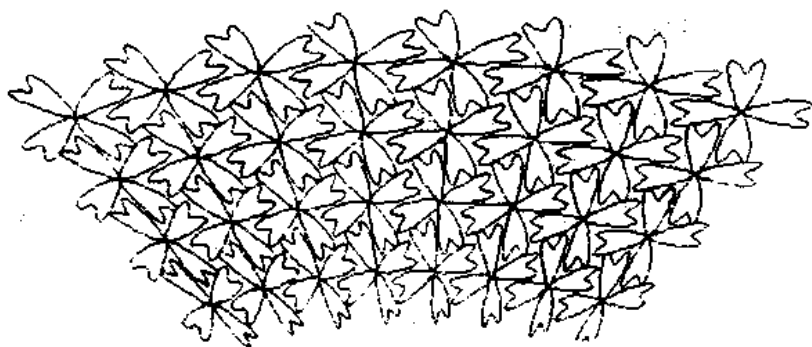


图 2.23

2.7 环形变换

将图 2.24 所示的带状图形变换为右下所示环状图形的变换，称为环状变换。环状变换将带状图的左右两端接在一起，因而带状图的上部被拉伸，图形下部被压缩。

带状图形上任意点 $P(X_1, Y_1)$ 变换为环状图上点 $P(X_2, Y_2)$ ，其坐标之间关系为：

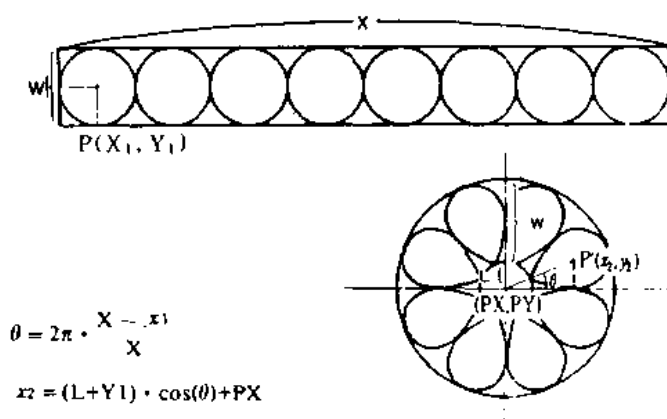
$$X_2 = (L + Y_1) * \cos(\text{TH}) + \text{PX}$$

$$Y_2 = (L + Y_1) * \sin(\text{TH}) + \text{PY}$$

式中

$$\text{TH} = 2\pi * (X - X_1) / X$$

L 为环形内径的 $1/2$, L 的值越小, 变换后图案的变形也越大。变换时, 应尽量多取原图内的数据点, 即使直线也要多分一些点, 否则变换效果不佳。



$$\theta = 2\pi * \frac{X - X_1}{X}$$

$$x_2 = (L + Y_1) * \cos(\theta) + \text{PX}$$

$$y_2 = (L + Y_1) * \sin(\theta) + \text{PY}$$

图 2.24

例 2-16 将 8 个并排的六瓣花型图案变换成环形的程序

```
100 ' pc 2-16
110 SCREEN 2 :CLS
120 PI = 3.14159:W = 150:X = 1200:I = 50
130 FOR PX = W / 2 TO 7 * W + W / 2 STEP W
140 FOR A = 0 TO 2 * PI STEP PI / 64
```

```

150 D=W/2*(4/5+1/5*SIN(18*A))*1.2
160 T=D*(1/2+1/2*SIN(6*A))
170 X1=PX+T*COS(A)
180 Y1=W/2-T*SIN(A)
190 GOSUB 260
200 IF A=0 THEN PSET (X2, Y2):BX=X2:BY=Y2
210 LINE-(X2, Y2)
220 NEXT A:LINE-(BX, BY)
230 NEXT PX
240 END
260 TH=2*PI*(X-X1)/X
270 X2=(L+Y1)*COS(TH)+320
280 Y2=(L+Y1)*SIN(TH)/2+100
290 RETURN

```

程序运行结果如图 2.25。

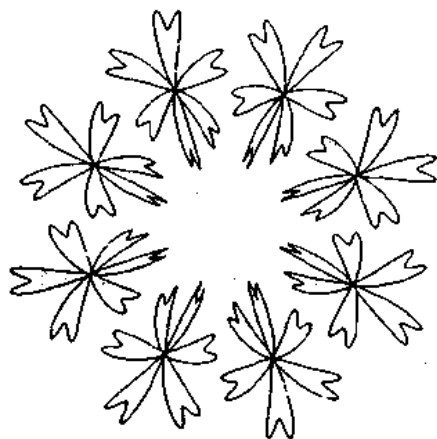


图 2.25

说明:

120 行 输入原图高 W、宽 X 和环半径 L;

130~230 行 计算原图的坐标 X_1 , Y_1 , 然后转入 250~290 行子程序进行变换。

例 2-17 二重环形变换程序例 1

```
100 pc 2-17
110 SCREEN 2 :CLS
120 DIM X(300), Y(300)
130 PI=3.14159:X=8000:W=200:L=0
140 GOSUB 520
150 FOR PY=W/4 TO 3*W/4 STEP W/2
160 FOR PX=X/16 TO X-X/16 STEP X/8
170 FOR I=1 TO J
180 X1=X(I)+PX:Y1=Y(I)+PY
190 GOSUB 310
200 IF I=1 THEN PSET (X2, Y2):BX=X2:BY=Y2
210 LINE-(X2, Y2)
220 NEXT I:LINE-(BX, BY)
230 NEXT PX, PY
240 END
310 TH=2*PI*(X-X1)/X
320 X2=(L+Y1)*COS(TH)+320
330 Y2=(L+Y1)*SIN(TH)/2+100
340 RETURN
520 FOR A=0 TO 2*PI STEP PI/120
530 J=J+1
```

```

540 D = 32 * (4 / 5 + 1 / 5 * SIN(18 * A))
550 T = D * (1 + SIN(6 * A))
560 X(I) = T * COS(A) * 1.2:Y(I) = T * SIN(A)
570 NEXT A
580 RETURN

```

程序运行结果如图 2.26.

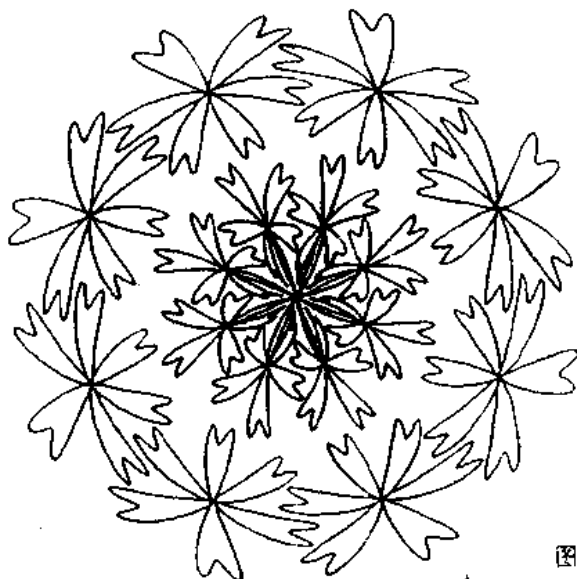


图 2.26

说明:

130 行 $L=0$ 环中心无空洞.

140 行 转入子程序 500, 计算原型图案的坐标.

150~230 行 画图, 由 190 行转入 310 行的子程序进行变换.

例 2-18 二重环形变换程序例 2

100 pc 2-18

```

110 SCREEN 2 :CLS
120 DIM X(300), Y(300)
130 PI=3.14159:X=800:W=200:L=0
140 GOSUB 520
150 FOR PY=W/4 TO 3*W/4 STEP W/2
170 FOR PX=X/16 TO X-X/16 STEP X/8
180 FOR I=1 TO J
185 X1=X(I)+PX:Y1=Y(I)+PY
190 GOSUB 310
200 IF I=1 THEN PSET(X2, Y2/2):BX=X2:BY=Y2/2
210 LINE-(X2, Y2/2)
220 NEXT I:LINE-(BX, BY)
230 NEXT PX, PY
240 END
310 TH=2*PI*(X-X1)/X
320 X2=(L+Y1)*COS(TH)+320
330 Y2=(L+Y1)*SIN(TH)+200
340 RETURN
520 FOR A=0 TO 2*PI STEP PI/80
530 J=J+1
540 D=15*SIN(8*(A+PI/16))
560 X(J)=D*COS(A)+45*COS(A)^3: Y(J)=D*SIN(A)+45*SIN(A)^3
570 NEXT A
580 RETURN

```

试运行程序并观察图形。将程序中 500 行以下的部分进行改动，可以画出许多美的图案。

例 2-19 二重变换花形程序例 3

```

100 'pc2-19
110 SCREEN 2 :CLS
120 DIM X(300), Y(300)
130 PI=3.14159:X=800:W=200:L=0
140 GOSUB 520
150 FOR PY=W/4 TO 3*W/4 STEP W/2
170 FOR PX=X/16 TO X-X/16 STEP X/8
180 FOR I=1 TO J
185 X1=X(I)+PX:Y1=Y(I)+PY
190 GOSUB 310
200 IF I=1 THEN PSET(X2, Y2/2):BX=X2:BY=Y2
210 LINE-(X2, Y2/2)
220 NEXT I:LINE-(BX, BY)
230 NEXT PX, PY
240 END
310 TH=2*PI*(X-X1)/X
320 X2=(L+Y1)*COS(TH)+320
330 Y2=(L+Y1)*SIN(TH)+200
340 RETURN
520 FOR A=0 TO 2*PI STEP PI/120
530 J=J+1
540 D1=40*SQR(ABS(COS(3*A)))
550 T=D1+10*SQR(ABS(COS(9*A)))
560 X(J)=T*COS(A)*1.2:Y(J)=T*SIN(A)
570 NEXT A

```


580 RETURN

则可得到如图 2.27 的图形。

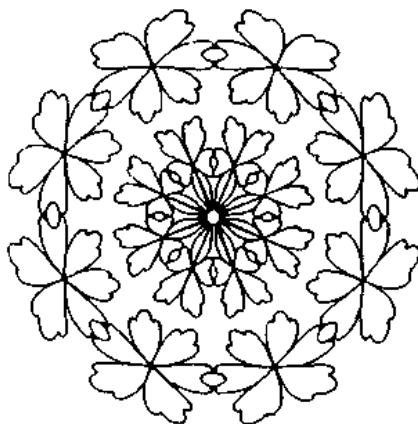


图 2.27

例 2-20 三重环状图案程序例

```
100 ' pc 2-20
110 SCREEN 2 :CLS
120 DIM X(241), Y(241)
130 PI=3.14159:W=198:X=660:L=0
140 GOSUB 510
150 FOR PX=W/6 TO X-W/6 STEP X/10
160 FOR PY=W/6 TO W-W/6 STEP W/3
170 FOR I=1 TO J
180 X1=X(I)+PX:Y1=Y(I)+PY
190 GOSUB 310
200 IF I=1 THEN PSET(X2, Y2):BX=X2:BY=Y2
```

```

210 LINE-(X2, Y2)
220 NEXT I:LINE-(BX, BY)
230 NEXT PX, PY
240 END
310 TH=2 * PI * (X-X1) / X
320 X2=(L+Y1) * COS(TH)+320
330 Y2=(L+Y1) * SIN(TH) / 2+100
340 RETURN
510 FOR A=0 TO 2 * PI STEP PI / 120
520 J=J+1
530 D=W / 6 * (4 / 5+1 / 5 * SIN(18 * A))
540 R=D * (1 / 2+1 / 2 * SIN(6 * A))
550 X(J)=1.4 * R * COS(A)
560 Y(J)=1.4 * R * SIN(A)
570 NEXT A
580 RETURN

```

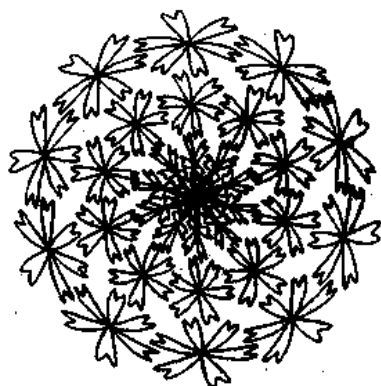


图 2.28

2.8 三角形变换

根据生息环境对生物的影响，可以推测生物的形态；变化支配生物形态的坐标系，将会生成新的形态。

例如，在图 2.29 中，当我们将位于正方形内的图案挤入一个三角形内时，图案将变形，生成新的图案。

设三角形为等边三角形，三角形高等于正方形的边长，则三角形边长 M 为：

$$M = 2 * L / \text{SQR}(3)$$

将正方形内图案的坐标值，全部平移到第 1 象限，即将正方形左下角顶点做为原点进行变换时，正方形内的点 $P_1(X_0, Y_0)$ 将变换为三角形内的点 $P_2(X_0 * Y_0 * M / L^2 + (L - Y_0) * M / L / 2, Y_0)$ ， P_2 点的 X 坐标发生了变化，其 Y 坐标不变。

采用这种点的变换而使图案产生形变生成新图案的方法，称为三角形变换。

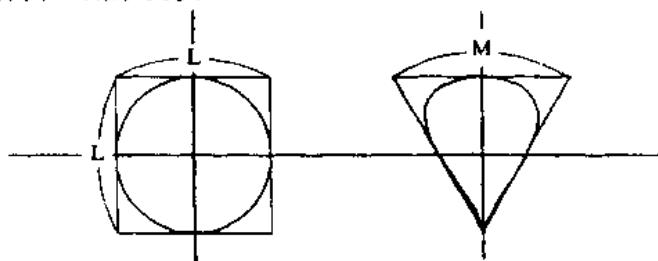


图 2.29

例 2-21 将正方形内 4 个六瓣花形进行三角形变换、再进行旋转生成新图案的程序。

100 pc2-21

```

110 SCREEN 2:CLS
120 PI=3.14159:L=180:M=2*L/SQR(3)
130 D=L/4*4.3
140 FOR AA=0 TO 5*PI/3 STEP PI/3
150   FOR PX=L/4 TO L/4+L/2 STEP L/2
160     FOR PY=L/4 TO L/4+L/2 STEP L/2
170       FOR A=0 TO 2*PI+PI/60 STEP PI/48
180         X0=D*(1/2+1/2*SIN(6*A))*COS(A)+PX
190         Y0=D*(1/2+1/2*SIN(6*A))*SIN(A)+PY
200         X2=X0*Y0*M/L^2+(L-Y0)*M/L/2-M/2
210         GOSUB 280
220         IF A=0 THEN PSET (X,Y/2)
230         LINE-(X,Y/2)
240       NEXT A
250     NEXT PY,PX,AA
260   END
280 X=X2*COS(AA)-Y0*SIN(AA)+320
290 Y=X2*SIN(AA)+Y0*COS(AA)+200
300 RETURN

```

说明:

120 行 设定 L 为正方形边长;

130 行 设定 D 为六瓣花形的半径;

140 行 循环语句为旋转循环, 步长 $\pi/3$, 将六个变换后的三角形图形填入正六边形内。

150~160 行 画正方形内图案的循环语句, 170 行的循环语句画六瓣花型, 由 180~190 行语句计算六瓣花型的坐标。

200 行 进行三角形变换，并在旋转后绘线。
程序运行结果如图 2.30。

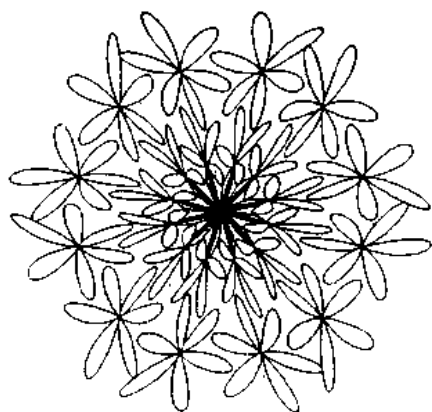


图 2.30

2.9 三角函数变换

如图 2.31 所示，在圆周上等间隔分布的点在直线上的

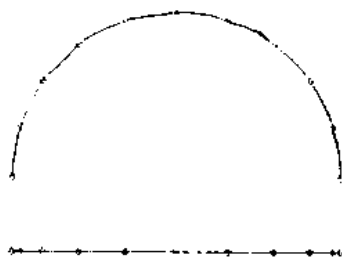


图 2.31

投影，不再是等间隔的点，这样，我们将原图放在圆周上进行假定投影从而使图形产生变换的方法，称为图形的三角函

数变换。这时，投影点是圆周上点的 X 分量。

例 2-22 将 6×6 个花型单元进行三角函数变换的程序

```
100 pc2-22
110 SCREEN 2 :CLS
120 DIM X(300), Y(300)
130 PI=3.14159:R=45
140 GOSUB 300
150 FOR PX=170 TO 470 STEP 60
160 FOR PY=50 TO 350 STEP 60
170 FOR I=1 TO J
180 X=X(I)+PX:Y=Y(I)+PY
190 GOSUB 260
200 IF I=1 THEN PSET(X2, Y2/2):BX=X2:BY=Y2/2
210 LINE-(X2, Y2/2)
220 NEXT I:LINE-(BX, BY)
230 NEXT PY, PX
240 END
260 X2=320-180 * COS(PI * (X-140) / 360)
270 Y2=200+180 * COS(PI * (X-20) / 360)
280 RETURN
300 FOR A=0 TO 2 * PI STEP PI / 60
310 J=J+1
320 D=R * (1+1 / 5 * SIN(12 * A))
330 T=D * (1 / 2+1 / 2 * SIN(4 * A))
340 X(J)=T * COS(A):Y(J)=T * SIN(A)
```

350 NEXT A:RETURN

说明:

300~350 行 花形单元子程序;

140~150 行 单元配置和绘图, 由 190 行转入变换子程序 (260~280 行)。

程序运行结果如图 2.32。

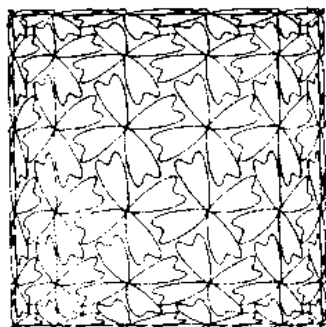


图 2.32

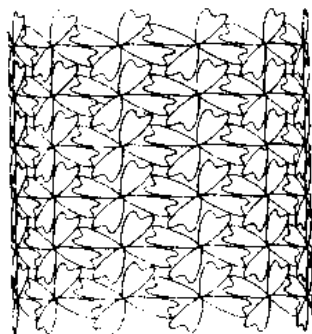


图 2.33

将程序中 270 行改为

270 Y2 = Y

则 Y 坐标不变, 运行结果如图 2.33.

同样, 若使 X 坐标不变, 可将 260 行改为:

260 X2 = X

试运行程序.

2.10 球面镜反射变换

为便于说明球面镜反射变换, 我们先看下面的程序例:

例 2-23: 球面镜反射变换程序

```
100 ' pc2-23
110 SCREEN 2:CLS
120 R = 150
130 FOR X = -180 TO 180 STEP 5
140 FOR Y = -180 TO 180 STEP 5
150 IF X^2+Y^2 < R^2 THEN GOSUB 200 ELSE XC=X:YC=Y
160 PSET(XC+320, YC / 2+100)
170 NEXT Y, X
180 END
200 IF X < 0 THEN S = -1 ELSE S = 1
210 IF X = 0 THEN X = 0.1
220 L = SQR(X^2+Y^2):BT = 2 * ATN(L / R)
230 TH = ATN(Y / X)
240 M = R * SIN(BT):XC = S * M * COS(TH)
250 YC = S * M * SIN(TH)
260 RETURN
```

程序运行结果如图 2.34.

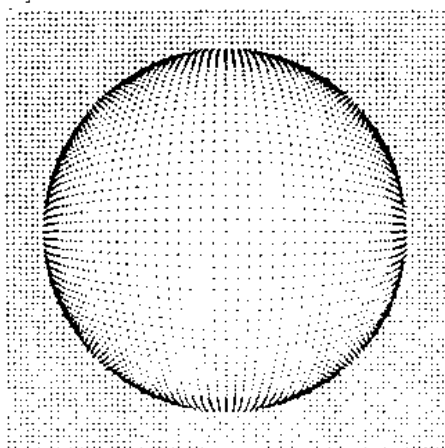


图 2.34

这个图形,如图 2.35 所示的那样,由 O 点射来的光,将球内的点 P_1 映为点 P_2 ,再把 P_2 投影到与 P_1 同一平面上绘出的。这样,平面上 P_2 点的坐标 $P_2(X_C, Y_C)$ 为:

$$XC = M \cdot \cos\theta$$

$$YC = M \cdot \sin\theta$$

式中,

$$M = R \sin\beta$$

$$\beta = 2\alpha = \tan^{-1} \frac{L}{R}$$

$$\theta = \tan^{-1} \frac{Y}{X}$$

$$L = \sqrt{X^2 + Y^2}$$

在本程序例中,上述变换是由 200~260 行的子程序完成的。

$$L = \sqrt{X^2 + Y^2}$$

$$\theta = \tan^{-1} \frac{Y}{X}$$

$$\beta = 2 \cdot \alpha = 2 \cdot \tan^{-1} \frac{L}{R}$$

$$M = R \sin \beta$$

$$\therefore X2 = M \cos \theta$$

$$Y2 = M \sin \theta$$

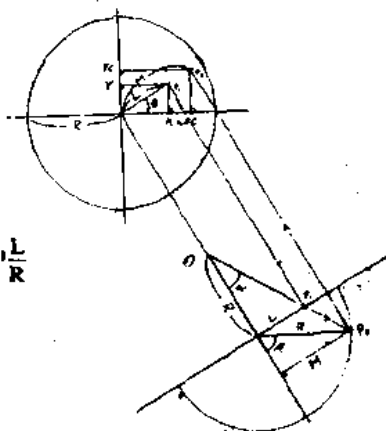


图 2.35

例 2-24 球面镜变换程序例

100 pc 2.24

110 SCREEN 2:CLS

120 DIM X(300), Y(300):PI=3.14159:R=150

130 INPUT UN

140 UV=360/UN:K=UV/2:SC=UV/100

150 GOSUB 510

160 FOR PX=-180+K TO 180-K STEP UV

170 FOR PY=-90+K TO 90-K STEP UV

180 FOR I=1 TO J

190 X1=X(I)*SC+PX:Y1=Y(I)*SC+PY

200 SQ=X1^2+Y1^2

```

210 IF SQ<R^2 THEN GOSUB 290 ELSE X=X1:Y=Y1
220 X=X+320:Y=Y/2+100
230 IF I=1 THEN PSET(X,Y):BX=X:BY=Y
240 LINE-(X,Y)
250 NEXT I:LINE-(BX,BY)
260 NEXT PY,PX
270 END

290 IF X1<0 THEN S=-1 ELSE S=1
300 IF X1=0 THEN X1=0.1
310 L=SQR(SQ):BT=2*ATN(L/R)
320 TH=ATN(Y1/X1)
330 M=R*SIN(BT):X=S*M*COS(TH):Y=S*M*SIN(TH)
340 RETURN

510 FOR A=0 TO 2*PI STEP PI/60
520 J=J+1
530 D=90*(4/5+1/5*SIN(12*A))
540 DD=D*(1/2+1/2*SIN(4*A))
550 X(J)=DD*COS(A):Y(J)=DD*SIN(A)
560 NEXT A:RETURN

```

程序中，130行 输入花型单元数，

140行 设定单元配置常数；

当输入不同的单元数 UV 及单元花型，可以画出多种不同的图案，如图 2.36 所示。

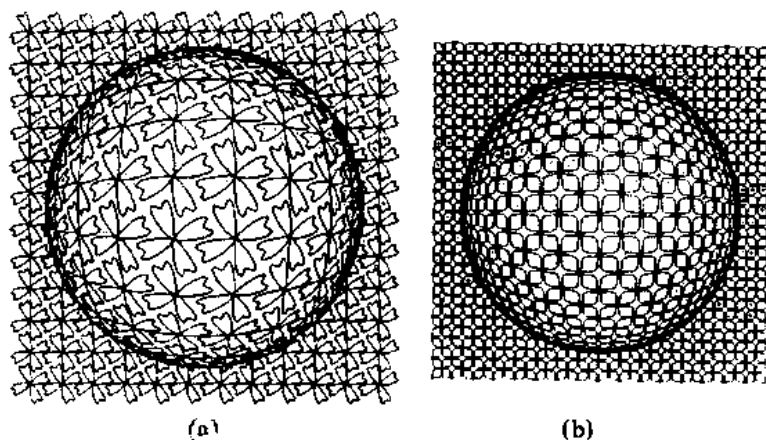


图 2.36

2.11 窗口——视见变换

在大于 2.00 版 BASICA 语言中，提供了窗口语句与视图口语句，用于实现由世界坐标系到屏幕坐标系的坐标变换，图形剪裁及选择图形在屏幕上的显示位置。

一、视图口语句 VIEW

VIEW 语句决定图形在屏幕上的显示位置，并可将显示屏幕再细分为若干个小的显示画面区域。

VIEW 语句格式：

VIEW SCREEN [(X1,Y1)-(X2,Y2)]颜色代码, 边框色码

说明：

参数(X1, Y1)-(X2, Y2)确定视见区左上角坐标和右下角坐标。必须取屏幕有效界限内的整数值；

颜色代码取值 0~3，选择视见区底色。缺省时，视见区不填颜色。

视见区彩色边框码取值 0~3, 缺省时不画边框;

SCREEN 是任选项, 缺省时所有绘图点与视见区有关。例如,

```
10 VIEW (10,10)-(200,100)
```

```
20 PSET (0,0)
```

执行后, 在屏幕上 (10, 10) 处显示一个点。

如果含有 SCREEN, 所有绘制点坐标都为绝对坐标, 只有在视见区以内的点才显示。例如,

```
10 VIEW SCREEN (10,10)-(200,100)
```

```
20 PSET (0,0)
```

执行后屏幕上不显示点, 而当取

```
20 PSET (10,10)
```

时, 才会在视见区内显示一点。

VIEW 语句在执行时有自动排序功能, 例如,

```
10 VIEW (100,100)-(5,0)
```

被自动排序为:

```
10 VIEW (5,0)-(100,100)
```

VIEW 语句可以确定多个视见区, 但在任何时刻仅有一个视见区是活动的。执行 RUN 和 SCREEN 语句将清除这些视见区。

注意, 当使用 VIEW 语句时, CLS 语句仅能清除当前视见区。为了清除整个屏幕, 须用 VIEW 语句清除所有的视见区, 然后用 CLS 语句清除屏幕。

例 2-25 确定四个视见区及显示图形

```
5 ' pc2-25
```

```
10 SCREEN 1,0:VIEW:CLS:KEY OFF
```

```

20 VIEW (1, 1)-(151, 91),, 1
30 VIEW (165, 1)-(315, 91),, 2
40 VIEW (1, 100)-(151, 195),, 3
50 VIEW (165, 105)-(315, 195),, 1
60 LOCATE 2, 4:PRINT"viewport"
70 LOCATE 2, 25:PRINT"viewport 2"
80 LOCATE 15, 4:PRINT"viewport 3"
90 LOCATE 15, 25:PRINT"viewport 4"
100 VIEW SCREEN (1, 1)-(151, 91):GOSUB 1000
110 VIEW SCREEN (165, 1)-(315, 91):GOSUB 2000
130 VIEW SCREEN (1, 100)-(151, 195):GOSUB 3000
140 VIEW SCREEN (165, 105)-(315, 195):GOSUB 4000
150 END
1000 CIRCLE (65, 60), 30, 2
1010 RETURN
2000 LINE (185, 50)-(290, 75), 1, B
2010 RETURN
3000 LINE (30, 150)-(100, 190), 2
3010 RETURN
4000 PSET (190, 150):DRAW"e15;f15;l30"
4010 RETURN

```

二、窗口语句 WINDOW

IBM PC BASIC 2.0以上版本在图形方式可以用 WINDOW 语句自定义坐标。WINDOW 语句的格式如下:

WINDOW SCREEN (X1, Y1)-(X2, Y2)

说明:

参数(X1, Y1),(X2, Y2)为世界坐标系的坐标, 它表示矩形窗口左上角顶点坐标和右下角顶点坐标, 当省略 SCREEN 项时, 则表示左下角和右上角的顶点坐标, 一般取值不受屏幕大小限制。

SCREEN 是任选项, 含有时表示世界坐标系与屏幕坐标系取向相同; 缺省时, 则在屏幕看见的是真正的世界坐标系坐标 (即两个坐标系 X 轴方向取向相同, 但 Y 轴方向取向相反)。

WINDOW 语句具有图形“剪裁”功能, 即在视见区内只显示窗口内的图形而不显示窗口外的图形。

WINDOW 语句可以对图形进行“放大”, 若窗口大于图象, 则图象被缩小; 若窗口小于图象, 在同样的视图口下, 图象被“放大”。

同样, WINDOW 语句具有自动排序功能, 例如,

```
10 WINDOW (100, 100)-(5, 5)
```

将自动排为:

```
10 WINDOW (5, 5)-(100, 100)
```

例 2-26 利用 WINDOW 语句“放大”图形

```
5 pc2-26
```

```
10 SCREEN1:CLS
```

```
20 WINDOW (-26, -26)-(26, 26)
```

```
30 CIRCLE(4, 4), 5, 1
```

```
40 WINDOW (-100, -100)-(100, 100)
```

```
50 CIRCLE(4, 4), 5, 1
```

```
60 END
```

程序自动选择整个屏幕为视见区。10~30行在屏幕上显示一个大圆；40行语句重新定义窗口，由50行语句画圆，但在屏幕上显示一个小圆。

例 2-27 用 WINDOW 进行图形变换程序

```
10  pc2-27
20  KEY OFF:CLSA:SCREEN 1,0
30  GOTO 160
50
70  LINE (X,0)-(-X,0),,, &HAA00
80  LINE (0,X)-(0,-X),,, &HAA00
100 CIRCLE (X/2,X/2),R
110 FOR P=1 TO 50:NEXT P
130 RETURN
140 ' =====
160 X=1000:WINDOW (-X,-X)-(X,X):R=20
180 GOSUB 50:FOR P=1 TO 1000:NEXT P:CLS
200 X=60:WINDOW (-X,-X)-(X,X):R=20
220 GOSUB 50:FOR P=1 TO 1000:NEXT:CLS
240 X=100:WINDOW (-5,-5)-(X,X):R=20
260 GOSUB 50:FOR P=1 TO 1000:NEXT:CLS
290 CLS:T=-50:U=100:X=U
300 FOR P=7 TO 1500:NEXT P
310 FOR I=1 TO 45
320 T=T+1:U=U-1:X=X-1:R=20
330 WINDOW (T,T)-(U,U):CLS:GOSUB 50
340 NEXT I
-106-
```


350 END

三、利用 WINDOW 和 VIEW 语句绘图

从上述中我们看到，窗口决定人们想看到的“世界”或景物”的那一部分，即在屏幕上要显示什么内容；视图口则决定在屏幕上的那一个区域显示出图形。一般而言，窗口多在世界坐标系内指定，视图口在点坐标系内指定，由 BASIC 语言自动完成由世界坐标系到点坐标系的坐标变换。

例 2-28 在世界坐标系给出图形数据的图形变换

```
5  pc2-28
10 READ PX1,PX2,PX3,PY1,PY2,PY3
20 GOSUB 100
30 FOR J=0 TO 100:NEXT J
40 TX=2:TY=3
50 GOSUB 200
60 GOSUB 100
70 END
100
110 SCREEN 1,0:CLS
120 WINDOW (0,0)-(6,6)
130 VIEW (0,0)-(319,199)
140 LINE(PX1,PY1)-(PX2,PY2),I
150 LINE-(PX3,PY3),I
160 LINE-(PX1,PY1),I
170 RETURN
200
210 PX1=PX1+TX:PY1=PY1+TY
```

```

220 PX2=PX2+YX:PY2=PY2+TY
230 PX3=PX3+TX:PY3=PY3+TY
240 RETURN
300 DATA 1,2,1,3,1

```

例 2-29 放大变换程序

```

5  .pc2-29
10 READ PX1,PX2,PX3,PY1,PY2,PY3
20 GOSUB 100
30 FOR J=0 TO 100:NEXT J
40 SX=0.5:SY=0.5
50 GOSUB 200
60 GOSUB 100
70 END

100
110 SCREEN 1,0:CLS
120 WINDOW (0,0)-(6,6)
130 VIEW (0,0)-(319,199)
140 LINE(PX1,PY1)-(PX2,PY2),1
150 LINE-(PX3,PY3),1
160 LINE-(PX1,PY1),1
170 RETURN

200
210 PX1=PX1*SX:PY1=PY1*SY
220 PX2=PX2*SX:PY2=PY2*SY
230 PX3=PX3*SX:PY3=PY3*SY
240 RETURN

```

图形数据由世界坐标系内给出。

2.12 图案的磁盘存取及打印

一、内存分配

IBM PC 内存总容量为 1MB。其中 ROM 为 40KB，可扩充到 256KB；RAM 可扩充到 620KB。内存空间分配如图 2.37。单色显示器和彩色显示器的缓冲区均在从 A0000H 开始的 128KB 范围内，其中单色显示器缓冲区为 4KB 地址为 B0000H 到 B0FFFH，彩色显示缓冲区 16KB 地址为 B8000H~BBFFFH。

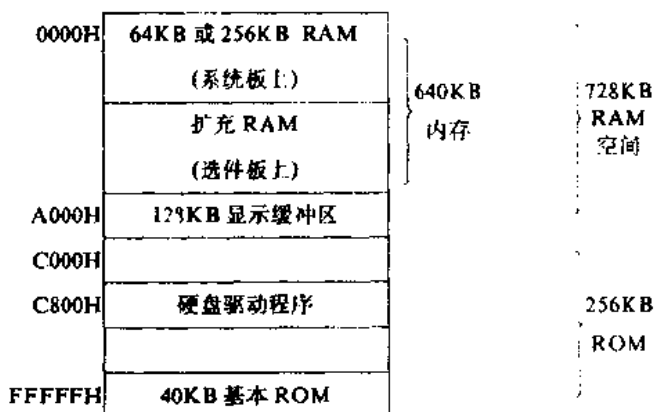


图 2.37

内存储器按地址可分为若干段。段的划分（用起始地址表示）由四个段寄存器 CS、SS、DS、FS 给出，它们分别为程序代码段、堆栈段、数据段和附加数据段。每段最大可达 64KB。当起动 BASIC 时，内存段的初始设置是 BASIC

的数据段 (DS)。BASIC 数据段是用户在内存工作空间的起始点。

二、显示缓冲区与屏幕扫描线的对应关系

单色显示器通常做为控制字符显示器。单色显示器的缓冲区包括字符存储器和属性存储器，存储量为 2KB，布局如图 2.38 所示。

	偶数字节地址	奇数字节地址	
B000	第 1 个字符	第 1 个字符属性	B001
	第 2 个字符	第 2 个字符属性	B003
B002	第 3 个字符	第 3 个字符属性	B005
B004			
B0F9E	第 2000 个字符	第 2000 个字符属性	B0F9F

图 2.38

彩色图形显示器的缓冲区又称为图形刷新存储器，容量 16KB。高分辨率图形的一个象元在内存中占用一个二进位，中分辨率时则占用两个二进位，文本方式一个象元占用 4 个二进位。屏幕象元位置与显示缓冲信息的对应关系如图 2.39。

B8000H		8000 字节偶数扫描线
B9F3FH		(0,2,4,...,198)
BA000H	不使用	
BBF3FH		8000 字节奇数扫描线
BBFFFH	不使用	(1,3,5,...,199)

图 2.39

三、屏幕图形信息的磁盘存储

屏幕图案信息能用 BSAVE 命令生成二进制磁盘文件，也能用 BLOAD 命令将这些二进制文件装入内存显示缓冲区，在屏幕上重新显示文字或图案。在此操作之前，必须用 DEF SEG 语句设置内存的段地址，BSAVE、BLOAD 等操作的实际地址为相对内存段地址的偏移量。

段地址定义语句的格式为

DEF SEG = AD

AD 取值在 0~65535 之间的数值表达式。

例如：

100 DEF SEG，将内存段设置为 BASIC 数据段（与启动 BASIC 时相同）；

200 DEF SEG = &HB800，设置彩色图形控制器的缓冲区的绝对地址 B8000；

注意：DEF SEG = &HB800 中 16 进制地址的最后一位数被丢弃。

表 2-1 列出了配有彩色显示器时，屏幕与磁盘间文件存取程序设计要点。

例如，若将中分辨率图形方式的屏幕图形信息存入磁盘时，应选择第 4 栏目的命令。首先用 DEF SEG = &HB800 选定屏幕缓冲区开始段地址为 B8000（16 进制），然后用 BSAVE F\$, 0, &H4000 将屏幕信息存入磁盘。F\$ 为文件名；最后用 DEF SEG 将内存段地址还原为正常值。

2

表 2-1

内容 \ 屏幕	IBM PC 单色显示	彩色图形控制器			
		40 列 本文方式	80 列 本文方式	中分辨率 图形方式	高分辨率 图形方式
设定内存 段地址	DEF SEG =& HB800	DEF SEG =& HB800	DEF SEG =& HB800	DEF SEG =& HB800	DEF SEG =& HB000
屏幕的 磁盘存取	BSAVE F\$, 0,&H1000	BSAVE F\$, 0,&H800	BSAVE F\$, 0,&H1000	BSAVE F\$, 0,&H4000	BSAVE F\$, 0,&H4000
从磁盘 调入内存	BLOAD F\$,0	BLOAD F\$,0	BLOAD F\$,0	BLOAD F\$,0	BLOAD F\$,0
还原内存 段地址	DEF SEG	DEF SEG	DEF SEG	DEF SEG	DEF SEG

例 2-30

```

1  pc2-30
10 PRINT"INPUT NO=1 OR 2"
20 INPUT NO
30 SCREEN 1:COLOR 0,0:CLS:KEY OFF
40 C=2: CX=160: CY=100
50 FOR R=20 TO 80 STEP 20
60 PSET (CX,CY), C
70 FOR T=0 TO 6.28318 STEP 1/R
80 IF NO=1 THEN R1=R*SIN(2*T):GOTO 100
90 R1=R*SIN(4*T)
100 X=CX+R1*COS(T):Y=CY+R1*SIN(T)
110 LINE-(X,Y), C

```

```

120 NEXT T, R
130 IF INKEY$ = "" THEN 130
140 DEF SEG = &HB800
150 BSAVE "PC2-30.PIC", 0, &H4000
160 DEF SEG
170 END

```

10~130 行 画出极坐标曲线图形;

140 行 设定内存段地址为 B8000 (16 进制);

150 行 以文件名 "PC2-30.PIC" 将屏幕信息存入磁盘;

160 行 还原内存段地址为正常值。

将磁盘装载的屏幕信息文件送入显示缓冲区并进行显示的程序

例 2-31

```

1  ' pc2-31
10 INPUT "FILENAME="; F$
20 SCREEN 1
30 DEF SEG = &HB800
40 BLOAD F$, 0
50 DEF SEG
60 IF INKEY$ "" THEN 60
70 END

```

10 行 输入磁盘文件名;

20 行 设定为中分辨率图形方式, 此设定应与磁盘文件的图形方式一致;

30 行 设定内存段地址;

40 行 装载磁盘文件, 显示图形;

50 行 还原段地址。

四、屏幕图案打印

屏幕图形打印是图案输出方法之一。这里介绍两种常用的打印方法。

(1) 用 GRAPHICS.COM 是 MS-DOS 系统的命令文件。操作步骤如下:

开启打印机电源, MS-DOS 引导成功后, 屏幕显示提示符 A> (或 C>) 后, 键入:

```
A> GRAPHICS
```

```
A> BASICA
```

按回车键, 屏幕显示 OK 标志, 装入并执行 BASIC 绘图程序, 屏幕显示图形后, 同时按 SHIFT 和 Prtsc 键, 打印机便打印出屏幕显示图形。

GRAPHICS 文件只适用于 FX-100, FAX-100, MBE-420 等图形打印机。

(2) 用 2024p、ALL9p、ALL24p 文件打印图形。

如果配用 M-2024 打印机时, 则首先在 CCDOS 或 MS-DOS 引导后, 调入 2024p 打印配接文件及 BASICA, 在装入并运行图形程序后, 按 SHIFT 和 Prtsc 两键, 即可打印出屏幕上的图形: 其他种类的打印机请调用配套的打印配接文件

2.13 图案的绘图机输出

利用打印机硬拷贝的屏幕图形, 往往比较粗糙, 图形的大小比例也无法控制。因而, 常用绘图机绘出要求精度较高的图形。

自动绘图机的种类很多，按工作方式可分为两类：一类是平台绘图机，另一类是滚筒式绘图机。平台绘图机精度高，速度快，造价比较昂贵；滚筒式绘图机精度和速度稍差，但价格比较便宜。

一、自动绘图机的结构。

图 2.40 是国内常用 SXY-880 型小型平台式绘图机示意图。其主要性能指标如下：

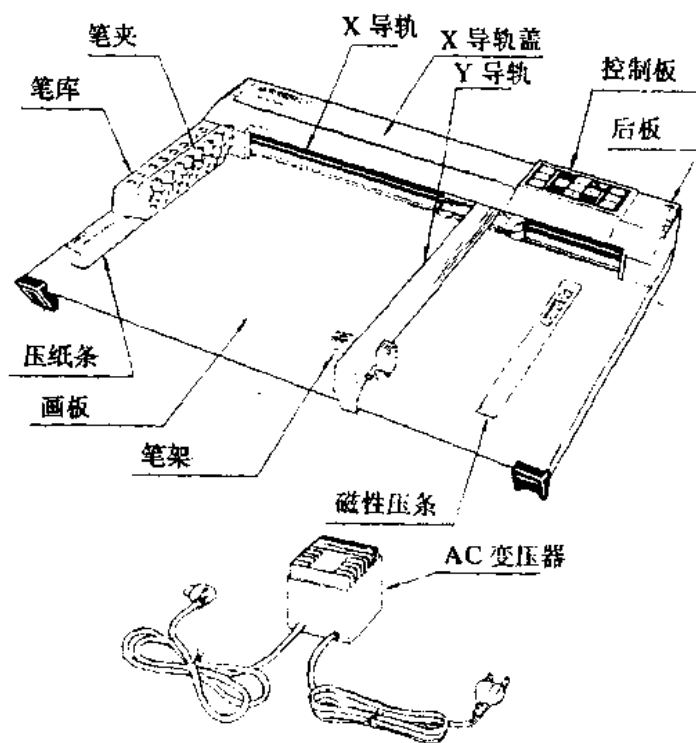


图 2.40

最大幅画: $380 \times 270\text{mm}^2$

绘图笔数: 8 支

分辨率: 0.05mm

最大速度: $200\text{mm}/\text{Sec}$

DXY-880 型绘图机主要由以下几部分组成:

(1) 绘图机台面 可以平放或斜放, 图纸由磁性压条压牢;

(2) 导轨 有 X 方向和 Y 方向两组。沿台面长度方向为 X 向导轨, 它固定在绘图机的两侧; 沿台面宽度方向为 Y 向导轨, 固定在可沿 X 向移动的横梁上, 笔架可在 Y 向导轨上移动。

(3) 传动机构 笔架和横梁的移动是由各自的驱动电机经过一套机械传动机构来带动的。

(4) 笔架 用于安装绘图笔。

二、工作原理

绘图机是绘图笔与图纸做相对运动的系统。绘图笔一般以直角坐标原点为起点, 沿两个坐标轴有四个基本方向运动, 即 $+X$ 、 $-X$ 、 $+Y$ 、 $-Y$ (如图 2.41)。由这四个方向又可组成沿 45° 和 135° 斜线运动方向, 即 $+X$ 、 $+Y$ 、 $+X$ 、 $-Y$ 、 $-X$ 、 $+Y$ 、和 $-X$ 、 $-Y$ 。此外绘图笔还有抬笔和落笔两个动作。自动绘图机绘图时, 绘图笔按照计算机及插补器送来的电脉冲指挥绘图笔按规定的方向运动。一个脉冲使绘图笔移动的距离称为步距。

绘图机只能接受十个基本指令, 即四个基本运动方向指令, 四个组合运动方向指令及抬笔和落笔指令。表 2-2 为本森 (Benson) 绘图机指令代码。

表 2-2

指令代码	画笔移动方向	指令代码	画笔移动方向
1	+x	9	+x, -y
2	-x	6	-x, +y
4	+y	10	-x, -y
8	-y	3	抬笔
5	+x+y	7	落笔

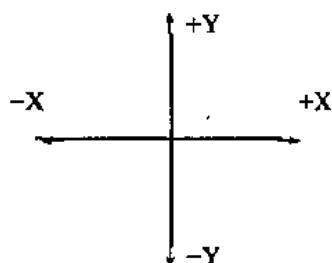


图 2.41

三、绘图机软件

利用绘图机画图时有两种方法。一种是手工编程，一种是计算机辅助编程。计算机辅助编程实际上是给计算机配以适当的软件、程序员应用这些软件，只需编出简单的源程序，就可绘出比较复杂的图形。

Dxy-880 型绘图机的绘图软件是以命令的形式来调用。典型命令有：

1. 归零命令“H”

格式: LPRINT "H"

功能: 将笔抬起移到绘图机坐标原点 (0,0)。

这个命令可用于减少绘图机的误差。

2. 画线命令“D”

格式:

`LRRINT“Dx1, y1, x2, y2, …xn, yn”`

参数分布: $-16383 \sim +16383$

功能: 从笔当前位置到点 (x_1, y_1) 画直线, 再到点 (x_2, y_2) 画直线, 依次类推, 一直画到点 (x_n, y_n) 。

3. 移动命令“M”

格式:

`LPRINT“Mx,y”`

参数分布: $-16383 \sim +16383$

功能: 将笔抬起移到点 (x, y) 处。

4. 相对画线命令“I”

格式:

`LPRINT“IΔx1, Δy1, Δx2, Δy2, ……Δxn, Δyn”`

参数分布: $-16383 \sim +16383$

功能: 以画笔当前位置 (x, y) 按增量 $\Delta x_1, \Delta y_1$ 到点 $(x+\Delta x_1, y+\Delta y_1)$ 画直线, 然后从点 $(x+\Delta x_1, y+\Delta y_1)$ 到点 $(x+\Delta x_1+\Delta x_2, y+\Delta y_1+\Delta y_2)$ 画线, ……依次类推, 一直画到点 $(x+\Delta x_1+\Delta x_2+\dots+\Delta x_n, y+\Delta y_1+\Delta y_2+\dots+\Delta y_n)$

5. 相对移动命令“R”

格式:

`LPRINT“R Δx, Δy”`

参数分布: $-16383 \sim +16383$

功能: 将笔抬起, 从当前位置 (x, y) 按增量 $\Delta x, \Delta y$ 移到点 $(x+\Delta x, y+\Delta y)$ 处。

6. 画坐标轴命令“x”

格式:

LPRINT“xp, q, r”

参数分布: p: 0, 1

q: $-16383 \sim +16383$

r: $1 \sim +16383$

功能: 从笔的当前位置画平行于 x 方向或 y 向的坐标轴。当 $p=0$ 时画 y 向坐标轴; 当 $p=1$ 时画 x 向坐标轴。q 参数用于确定刻度间的距离; r 参数用于确定坐标轴上刻度的数目。

7. 画圆命令“C”

格式:

LPRINT“Cx, y, 01, 02”

参数分布: x, yr: $-16383 \sim +16383$

01, 02: $-32767^\circ \sim +32767^\circ$

功能: 以 x, y 为圆心, r 为半径画圆或圆弧。起始角为 01, 终止角为 02。当 $01 < 02$ 时按逆时针方向画; 当 $01 > 02$ 时按顺时针方向画。

8. 相对画圆命令“E”

格式:

LPRINT“Er, 01, 02”

参数分布: r: $-16383 \sim +16383$

01, 02: $-32767^\circ \sim +32767^\circ$

功能：以笔的当前位置为起点，以 r 为半径画圆或圆弧，该圆或圆弧的起始角为 θ_1 ，终止角为 θ_2 。当 $\theta_1 < \theta_2$ 时按逆时针方向画；当 $\theta_1 > \theta_2$ 时按顺时针方向画。

Dxy-800 型绘图机还配备了一些其它绘图命令：如画剖面线命令、换笔命令等。

四、程序例：

例（1）画一正方形。其坐标顶点为：

(10, 10), (100, 10), (100, 100), (10, 100)

程序如下：

```
10 LPRINT'M10, 10"  
20 LPRINT'D100, 10, 100, 100, 10, 100, 10, 10"  
30 LPRINT'H"  
40 END
```

例（2）画一个半圆弧和以圆弧直径为边长的正方形衔接组成的图案。

```
10 LPRINT'M10, 60"  
20 LPRINT'I0, -50, 50, 0, 0, 50"  
30 LPRINT'E25, 0, 180"  
40 LPRINT'H"  
50 END
```

第三章 图案程序 60 例

本章给出 60 个有实用价值的图案程序。改变程序中相应参数，又可生成许多新颖图案。

例 3-1 圆图案

```
100 'pc3-1
110 SCREEN 2: CLS
120 pi=3.14159: L=50
130 FOR AL=0 TO 2*PI STEP PI/24
140 X1=L * COS(AL) * 7/5
150 Y1=L * SIN(AL)
160 GOSUB 210
170 CIRCLE(X2, Y2), 50
180 NEXT
190 END
210 X2=X1+320
220 Y2=-Y1+100
230 RETURN
```

这是一个在半径 $L=50$ 的圆周上取等间隔点为圆心，以半径 $R=50$ 画圆的图案。

140~150 行 计算圆心的 X , Y 坐标；

210~220 行 确定圆心在屏幕上的位置，然后同 170 行画圆。程序运行结果如图 3.1

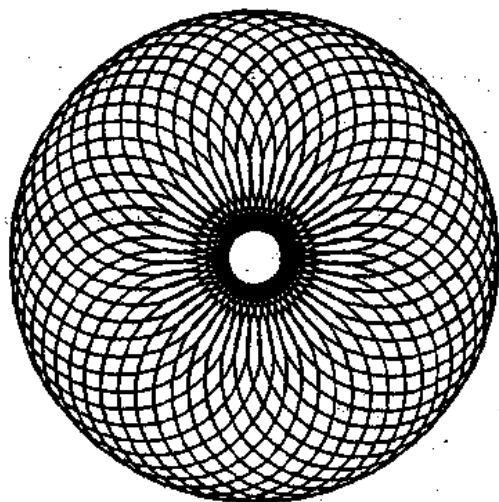


图 3.1

例 3-2 正弦波图案

```

100 'pc3-2
110 SCREEN 2: CLS
120 PI=3.14159
130 FOR PY=25 TO 175 STEP 4
140 FOR AL=0 TO 4*PI STEP PI/20
150 X=360/(4*PI)*AL+140
160 Y=PY-10*SIN(AL-2*PI/300*(PY-25))
170 IF AL=0 THEN PSET(X, Y)
180 LINE-(X, Y)
190 NEXT AL
200 NEXT PY
210 END

```


程序中横轴取角度值，纵轴取 ASIN(AL) 的值画出正弦函数的波形。

130 行每循环一次画一条正弦曲线，同时在 Y 方向上向下平移 4 点。

140 行以 AL 为变量画正弦曲线，由 150 行计算 X 的坐标值。

160 行括号中的 $2\pi / 300 * (\text{PY} - 25)$ ，当 $\text{PY} = 25$ 时，相位移为 0； $\text{PY} = 175$ 时，相位移为 2π 。程序运行结果如图 3.2。

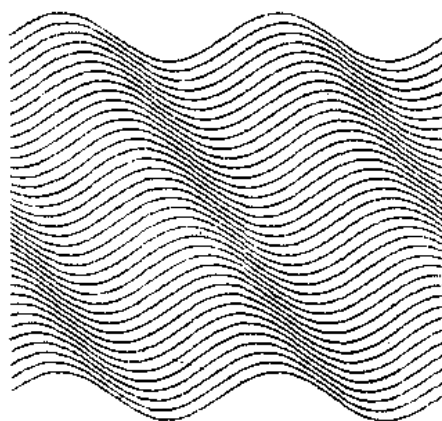


图 3.2

例 3-3 合成正弦函数图案

```
100 'pc3-3
110 SCREEN 2: CLS
120 PI=3.14159
130 FOR PY=25 TO 175 STEP 3
```

```

140 FOR AL=0 TO 4*PI+STET PI/40
150 X=360/(4*PI)*AL+140
155 P=10*SIN(2*AL)
160 Y=PY-P*SIN(AL-2*PI/150*(PY-25))
170 IF AL=0 THEN PSET(X, Y)
180 LINE -(X, Y)
190 NEXT AL
200 NEXT PY
210 END

```

这是由函数

$$y = 10 \times \sin(2x) \sin x$$

波形组成的图案。程序运行结果如图 3.3。

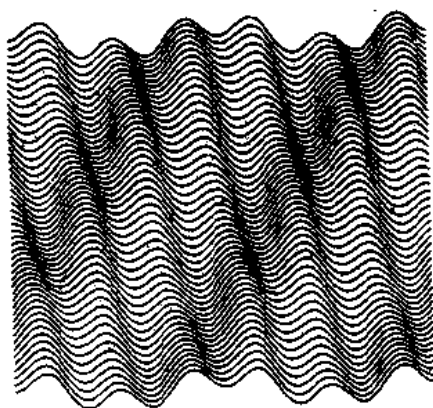


图 3.3

如果将程序中追加

```
157 P=P*(PY-25)/150
```

并将 160 行中的 $2*PI$ 改为 $4*PI$ ，则如图 3.4。

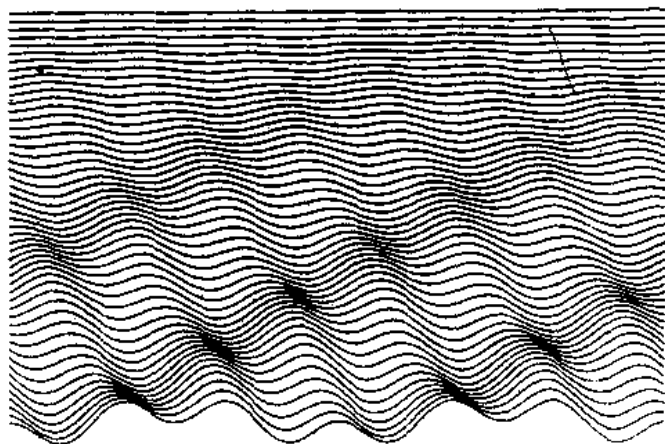


图 3.4

例 3-4 花卉 1

```

100 'pc3-4
110 SCREEN 2: CLS
120 PI=3.24259: T=6: R=.50
130 FOR A=0 TO 2*PI STEP PI/(20*T)
140 C=R*(2/3+1/3*SIN(I*A))
150 X=320+C*COS(A)*7/5
160 Y=100-C*SIN(A)
170 IF A=0 THEN PSET(X, Y): BX=X: BY=Y ELSE LI
    NE-(X,Y)
180 NEXT
190 LINE-(BX, BY)
200 END

```

这是一个圆和正弦波形迭加生成图案的程序。在半径 R 为 150 的圆上迭加振幅为 50 的正弦波时，如图 3.5。

若将程序中 140 行的 $2/3$ 和 $1/3$ 均改为 $1/2$ ，则程序运行结果如图 3.6。

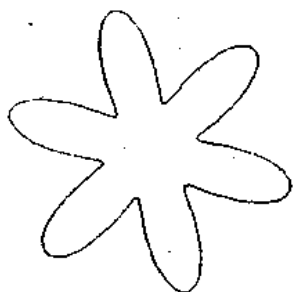


图 3.5

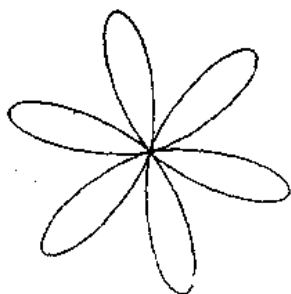


图 3.6

例 3-5 花卉 2

```

100 'pc3-5
110 SCREEN 2: CLS
120 PI=3.14159
130 INPUT T
140 D=100
150 FOR A=0 TO 2*PI STEP PI/(20*T)
160 B=D+D/5*SIN(3*T*A)
170 C=B*(1/2+1/2*SIN(T*A))
180 X=320+C*COS(A)*7/5
190 Y=100-C*SIN(A)
200 IF A=0 THEN PSET(X, Y): BX=X: BY=Y ELSE LI
    NE-(X, Y)

```

```

210 NEXT
220 LINE-(BX, BY)
230 END

```

程序运行结果如图 3.7 所示。这是一个圆和三次谐波迭加的图案。如果将程序 160 行中的 $D / 5 * \sin(3 * T * A)$ 改为 $D / 5 * \sin(4 * T * A)$ ，则程序运行结果是一个圆和四次谐波迭加的图案，如图 3.8。

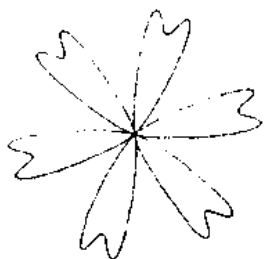


图 3.7

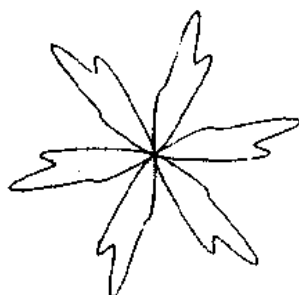


图 3.8

例 3-6 使用 RND 函数的图案

```

100 'pc3-6
105 SCREEN 2: CLS
110 FOR X=140 TO 500 STEP 15
120 FOR Y=25 TO 150 STEP 25
130 RX=X+8*RND(1)
140 PSET (RX, Y)
150 LINE -STEP(5*RND(3), 23)
160 NEXT Y, X
170 END

```

程序中由双重循环确定所画线段的起点坐标值。130 行用随机函数 $8 * \text{RND}(1)$ 使 X 坐标值在 0~8 的范围内变化。

画线终点的 Y 坐标的相对偏差值取 23, X 坐标的相对偏差值由 $5 * \text{RND}(1)$ 在 0~5 内随机取值。程序运行结果如图 3.9。

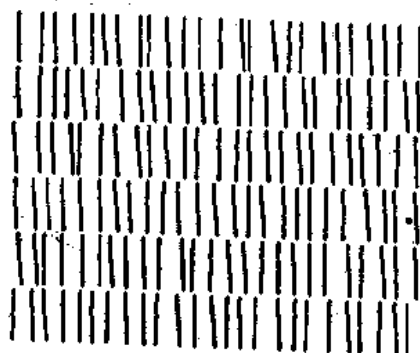


图 3.9

例 3-7 随机画圆图案

```
100 'pc3-7
110 SCREEN 2: CLS
120 LINE(140, 10)-(500, 190), ,B
130 FOR I=1 TO 200
140 X=500-360 * RND(1)
150 Y=190-180 * RND(2)
160 CIRCLE(X, Y), 5
170 NEXT
180 END
```

这是一个随机画圆的图案。圆心的坐标由 140~150 行

随机产生，由 160 行画半径为 5 的圆。程序运行如图 3.10。

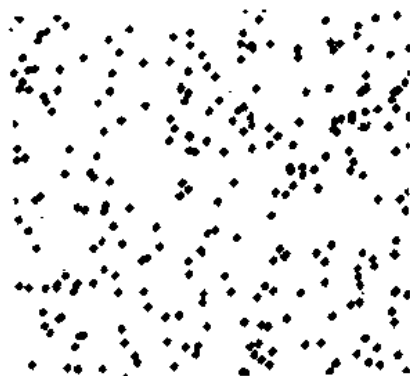


图 3.10

例 3-8 瓣数随机变化的花瓣图案

```
100 'pc3-8
110 SCREEN 2: CLS
120 PI=3.14159
130 D=45
140 FOR PX=185 TO 455 STEP 90
150 FOR PY=65 TO 335 STEP 90
160 T=10-INT(8 * RND(1))
170 FOR A=0 TO 2 * PI STEP PI / (10 * T)
180 B=D+D / 5 * SIN(3 * T * A)
190 C=B * (1 / 2+1 / 2 * SIN(T * A))
200 X=PX+C * COS(A)
210 Y=PY-C * SIN(A)
220 IF A=0 THEN PSET (X, Y / 2): BX=X: BY=Y / 2
230 LINE -(X, Y / 2)
240 NEXT: LINE-(BX, BY)
```

250 NEXT PY, PX

260 END

程序中利用例 3-4 的花型单元, 但花型瓣数由随机画数确定。

160 行中 $T = 10 - \text{INT}(8 * \text{RND}(1))$

产生随机变数, 使花瓣数在 3-10 范围内变化, 运行结果如图 3.11。

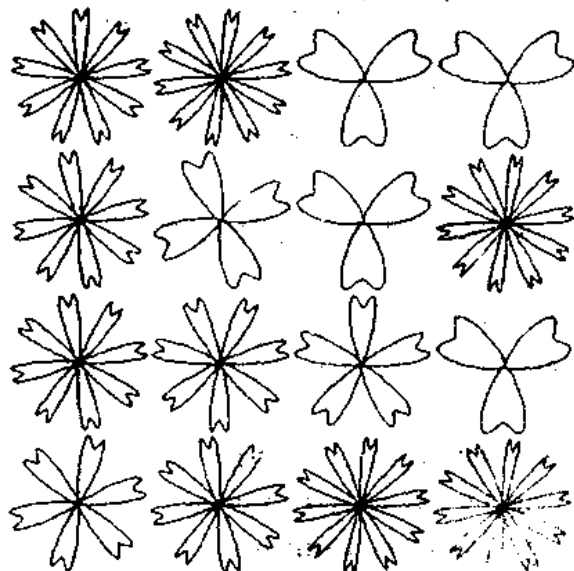


图 3.11

例 3-9 随机画 N 角形组成的图案

100 'pc3-9

110 SCREEN 2: CLS

120 PI=3.14159: N=2

130 FOR X=140 TO 500 STEP 20

140 FOR Y=10 TO 190 STEP 10

—130—


```

145 N=8-INT(6 * RND(1))
150 FOR A=0 TO 2 * PI STEP 2 * PI / N
160 XO=X+4 * SIN(A): YO=Y-4 * COS(A)
170 IF A=0 THEN PSET(XO, YO): BX=XO: BY=YO
180 LINE-(XO, YO)
190 NEXT A
200 LINE-(BX, BY)
220 NEXT Y, X
230 END
900 LINE-(BX, BY)

```

这是一个画随机 N 角形($N=3\sim 8$)并将其整齐排列的程序。运行结果如图 3.12。

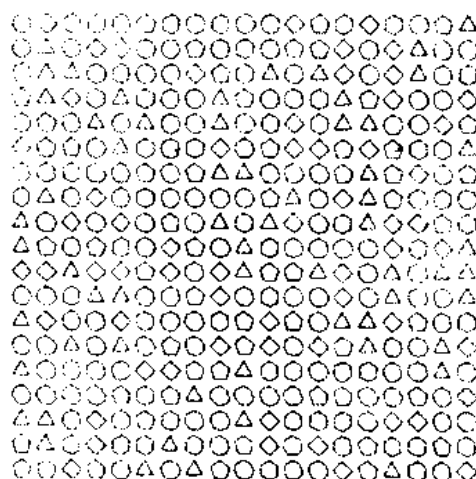


图 3.12

例 3-10 按 3~8 边形顺序排列的图案

例 3-11 按函数规律分布的图案

```
100 'pc3-11
110 SCREEN 2: CLS
120 PI=3.14159: NN=50: AM=40
130 DIM X(NN+1), Y(NN+1)
140 FOR M=1 TO NN+1
150 X(M)=140+360/NN*(M-1)
160 NEXT M
170 FOR M=1 TO NN+1
180 Y(M)=20+360/NN*(M-1)4190
190 NEXT M
200 FOR M=1 TO NN+1
210 FOR N=1 TO NN+1
220 L=AM*SIN(2*PI/360*(Y(N)-20))*SIN(PI/360*(X
(M)-140))
230 K=AM*SIN(2*PI/360*(Y(M)-20))*SIN(PI/360*(
X(N)-140))
240 PX=L+X(M)
250 PY=K/2+Y(N)/2
260 CIRCLE(PX, PY), 2
270 NEXT N, M
280 END
```

这是一个由点按一定规律排列构成图案的例子。这里的“点”实际上是一个“小圆”，如图 3.14。

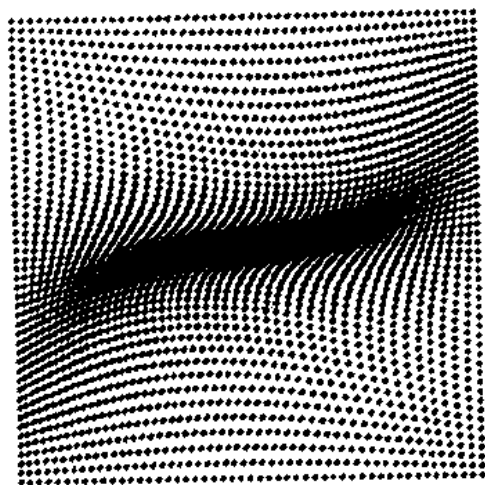


图 3.14

例 3-12 正弦函数规律分布点图案

```

100 'pc3-12
110 SCREEN 2: CLS
120 PI=3.14159
130 FOR PY=20 TO 380 STEP 5
140     B=4 * PI / 360 * (PY-20)
150 FOR A=0 TO 5 * PI STEP PI / 13
160     X=360 / (5 * PI) * A+140
170     Y=PY+25 * SIN(B) * SIN(A)
180     PSET (X, Y / 2)
190     NEXT A
200 NEXT PY
210 END

```

程序运行结果如图 3.15.

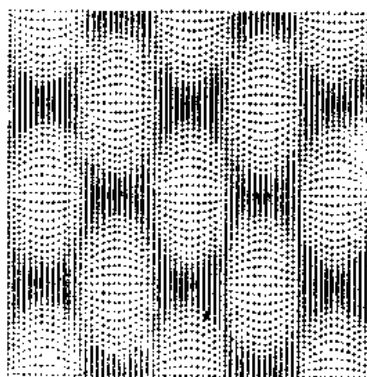


图 3.15

例 3-13 按三角函数变换规律分布的直线

```

100 'pc3-13
110 SCREEN 2: CLS
120 PI = 3.14159
130 FOR A = 0 TO PI STEP PI / 50
140 X = 320 - 180 * COS(A)
150 LINE(X, 10)-(X, 190)
210 NEXT A
220 END

```

程序运行结果如图 3.16。图案中各直线按三角函数变换规律分布。140 行语句计算各直线的 X 坐标。语句中的 $180 * \cos(A)$ 计算以宽度（360 点）为直径的上半圆上等间隔点在直径（横轴）上投影点的位置，使得图案具有立体感。

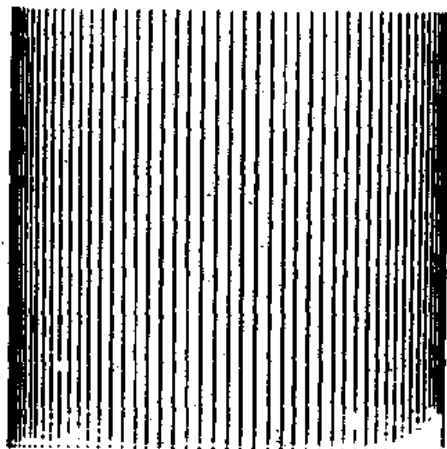


图 3.16

例 3-14 按对数规律分布的直线图案 1

```

100 'pc3-14
110 SCREEN 2
120 FOR N=1 TO 25
130 X=140+360*LOG(N)/LOG(25)
140 LINE(X, 10)-STEP (0, 90)
150 NEXT N
160 FOR N=1 TO 25
170 X=500-360*LOG(N)/LOG(25)
180 LINE (X, 100)-STEP(0, 90)
190 NEXT N
200 LINE(140, 10)-(500, 190), ,B
210 LINE(140, 100)-(500, 100)
230 END

```

这是一个按对数规律用直线分割平面所构成的图案。程

序运行结果如图 3.17.

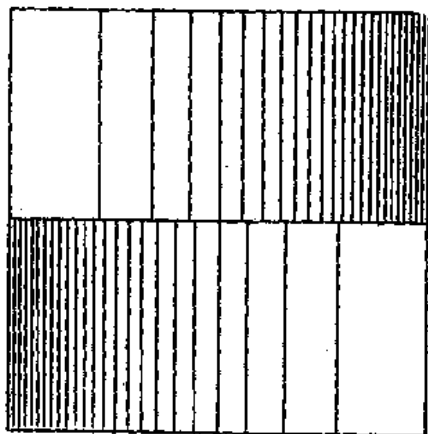


图 3.17

例 3-15 按对数规律分布的直线图案 2

```
100 'pc3-15
110 SCREEN 2
120 FOR J=1 TO 9
130 I=I+1: Y=10*I
140 FOR N=1 TO 50
150 X=140+360*LOG(N)/LOG(50)
160 LINE(X, Y)-STEP(0, 10)
170 NEXT N
180 I=I+1: Y=10*I
190 FOR N=1 TO 50
200 X=500-360*LOG(N)/LOG(50)
210 LINE(X, Y)-STEP(0, 10)
220 NEXT N
```

```

230 NEXT J
240 FOR Y=10 TO 190 STEP 10
250 LINE(140, Y)-SETP(360, 0)
260 NEXT Y
270 END

```

该程序是将例 3-14 中的直线压缩 20 倍，然后重复 9 次画出图案，如图 3.18。

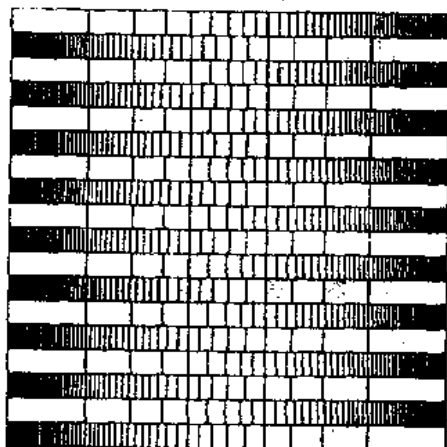


图 3.18

例 3-16 织物组织图案 1

```

100 'pc3-16
110 SCREEN 2: CLS
130 FOR PX=140 TO 470 STEP 60
140 I=I+1
150 FOR PY=10 TO 175 STEP 30
160 IF J=6 THEN J=1 ELSE J=J+1
170 IF (I MOD 2)≠(J MOD 2) THEN 230

```



```

180 FOR A=0 TO PI STEP PI/25
190 X=30-30 * COS(A)
200 LINE(PX+X, PY)-STEP(0, 30)
210 NEXT A
220 GOTO 270
230 FOR A=0 TO PI STEP PI/15
240 Y=30-30 * SIN(A)
250 LINE(PX, PY+Y)-STEP(60, 0)
260 NEXT A
270 NEXT PY, PX
280 END

```

程序运行结果如图 3.19.

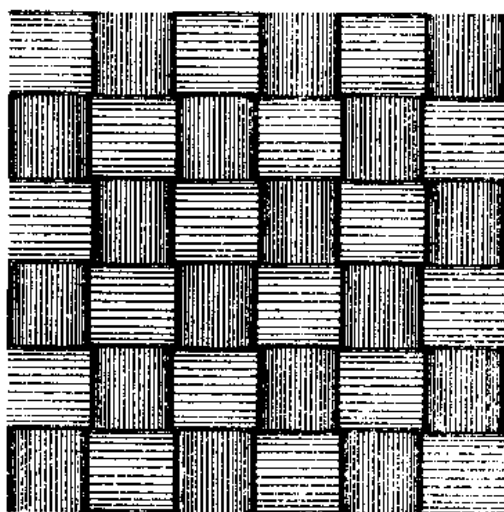


图 3.19

例 3-17 织物组织图案 2

```
100 'pc3-17
```

```

110 SCREEN 2: CLS
130 FOR PX=140 TO 470 STEP 60
140 IF I=6 THEN I=1 ELSE I=I+1
150 FOR PY=10 TO 175 STEP 30
160 IF J=6 THEN J=1 ELSE J=J+1
170 IF (I MOD 2)=(J MOD 2) THEN 230
180 FOR X=0 TO 60 STEP 3
200 LINE(PX+X, PY)-STEP(0, 30)
210 NEXT X
220 GOTO 270
230 FOR Y=0 TO STEP 3
250 LINE(PX, PY+Y)-STEP(60, 0)
260 NEXT Y
270 NEXT PY, PX
280 END

```

程序运行结果如图 3.20。

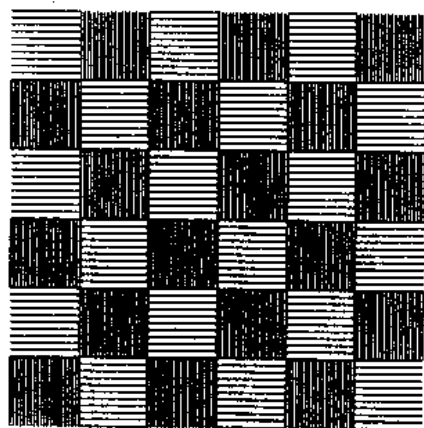


图 3.20

例 3-18 直线分割平面

```

100 'pc3-18
110 SCREEN 2: CLS
120 P= SQR(5): Q=(P+1)/2: R=(-P+1)/2: S=1/P
130 FOR N=1 TO 12
140 T=S*(QN-RN)
150 LINE(T+119, 0)-STEP(0, 199)
160 LINE(521-T, 0)-STEP(0, 199)
170 LINE (120, T-1)-STEP(400, 0)
190 NEXT
200 END

```

这是一种按费数纳奇数列分割平面的图案。数列的一般表达式为:

$$1, 1, 2, 3, 5, 8, 13, \dots, \frac{1}{\sqrt{5}} \left\{ \frac{(\sqrt{5}+1)^n}{2} - \frac{(-\sqrt{5}+1)^n}{2} \right\}$$

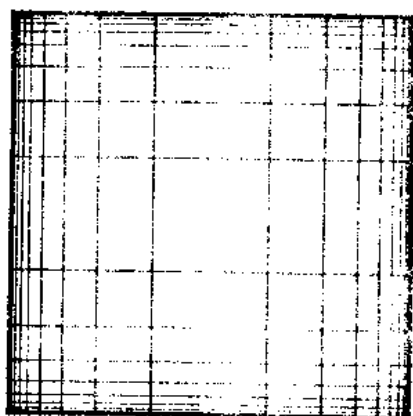


图 3.21

即前两项之和等于后一项。图 3.21 为从正方形各边向内部按其规律画线而形成的图案。

例 3-19 线段图案

使线段的端点按一定的函数规律变化、使之形成节奏感很强的图案。

```
100 'pc3-19
110 SCREEN 2: CLS
120 PI=3.14159
130 FOR A=0 TO PI STEP PI/380
140 PX=320: PY=100
150 X1=280 * COS(1.6 * A)+PX
160 Y1=199-((90 * SIN(8 * A)) * COS(A / 2.5))+PY
170 X2=280 * COS(1.8 * A)+PX
180 Y2=Y1
```

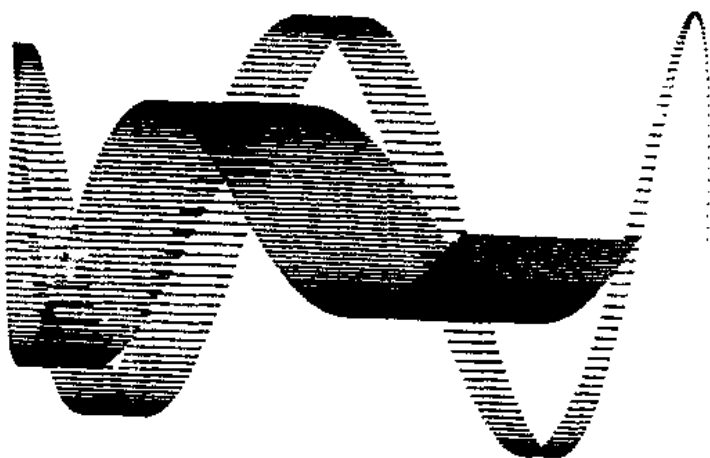


图 3.22

190 LINE(X1, Y1)-(X2, Y2)

200 NEXT A

210 END

当线段的端点按三角函数规律变化时, 程序运行结果如图 3.22.

例 3-20 矩形图案

100 'pc3-20

110 SCREEN 2: CLS

120 PI=3.14159: T=2

130 FOR A=0 TO PI STEP PI/240

140 PX=320: PY=100

150 X1=280 * COS(1.6 * A)+PX

160 Y1=199-((90 * SIN(8 * A)) * COS(A / 2.5)+PY)

170 X2=280 * COS(1.8 * A)+PX

180 Y2=Y1

190 GOSUB 230

200 NEXT A

210 END

230 AA=0

240 XX1=X1+T * COS(AA+PI / 2): YY1=Y1+T * SIN(AA+PI / 2)

250 XX2=X1+T * COS(AA+3 * PI / 2): YY2=Y1+T * SIN(AA+3 * PI / 2)

260 XX3=X2+T * COS(AA+PI / 2): YY3=Y2+T * SIN(AA+PI / 2)

270 XX4=X2+T * COS(AA+3 * PI / 2): YY4=Y2+T * SIN(AA+3 * PI / 2)

```

280 LINE(XX1, YY1)-(XX2, YY2)
290 LINE-(XX4, YY4): LINE-(XX3, YY3)
300 LINE-(XX1, YY1)
310 RETURN

```

将前例中的直线段改为矩形，并由 120 行语句中的 T 值控制矩形的宽度。当 $T=2$ 时，程序运行结果如图 3.23。



图 3.23

例 3-21 直线图案 1

```

100 'pc3-21
110 SCREEN 2: CLS
120 PI=3.141549
125 LINE(0, 0)-(639, 199), , B
130 FOR A=1.5 TO 2*PI STEP PI/120
140 X1=280+280*COS(.25*A)
150 Y1=100+(30-80*SIN(3*A))*COS(A/2.5)

```

```

160 X2=280+280 * COS(.25 * A+PI / 4)
170 Y2=100+(30-80 * SIN(3 * A-PI / 4)) * COS(A / 2.5)
180 LINE(X1, Y1)-(X2, Y2)
190 NEXT A
200 END

```

程序运行结果如图 3.24.

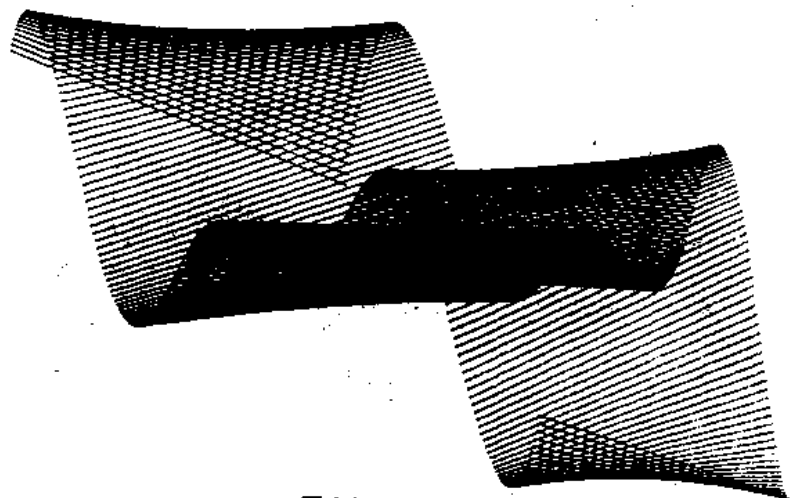


图 3.24

例 3-22 直线图案 2

```

100 'pc3-22
110 SCREEN 2: CLS
120 PI=3.14159
125 LINE(0, 0)-(639, 199), ,B
130 FOR A=0 TO 2 * PI STEP PI / 250
140 X0=320+250 * COS(A)
150 Y0=100+47 * SIN(5 * A) * COS(A / 1.5)

```

```

160 X1=88*SIN(A): Y1=28*SIN(A)
170 X2=X1: Y2=-Y1
180 LINE(X0, Y0)-STEP(X1, Y1)
190 LINE(X0, Y0)-STEP(X2, Y2)
200 NEXT A
210 END

```

程序执行结果如图 3.25。

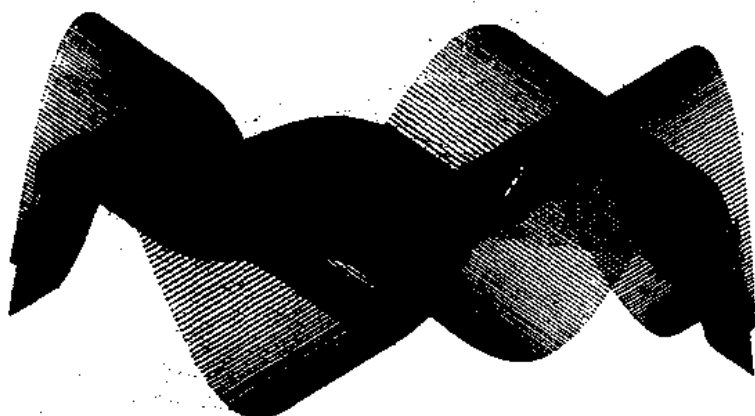


图 3.25

例 3-23 线段端点依函数规律分布的图案

```

100 'pc3-23
110 SCREEN 2: CLS
120 PI=3.14157: D=150: E=50
130 FOR A=0 TO 2*PI*(1+1/2)*COS(A)) * COS(A)
150 X1=250+1.25*L * COS(A)
160 M=E+E/3*(1+1/2*SIN(12*A))*COS(A)

```



```

170 X2=250 +1.25 * M * COS(A)
180 O=D+D / 3 * (1+1 / 2 * COS(10 * A)) * SIN(A)
190 P=E+E / 2 * (1+1 / 2 * COS(15 * A)) * SIN(A)
210 Y2=125-P * SIN(A) / 2
220 LINE (X1, Y1)-(X2, Y2)
230 NEXT A
240 END

```

用数学公式计算线段端点的坐标值，并在两端点间画线，可绘出多种图案。本程序按两个三角函数相乘所得函数的规律，由循环语句依次计算 (X1, Y1)、(X2, Y2) 的值，用 220 行语句画线生成图案，如图 3.26。变化 SIN、COS 后()内的数值，可绘出多种丰富多彩的图案。

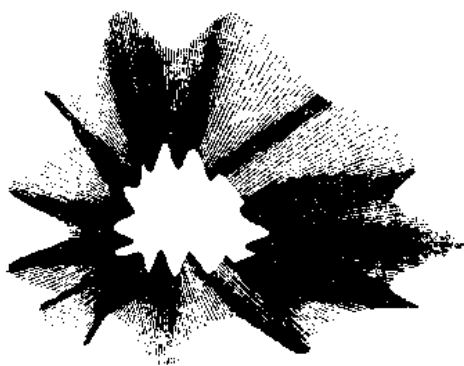


图 3.26

例 3-24 多层线段图案

```

100 'pc3-24
110 SCREEN 2: CLS

```

```

120 PX=320: PY=100
130 PI=3.14159
140 FOR A=0 TO 2*PI STEP PI/90
150 X1=300*SIN(A+PI/2)*3+PX
160 Y1=-90*COS(A+PI/2)*3+PY
170 X2=300*COS(A)+PY
180 Y2=150*COS(A)+PX
190 X3=150*COS(A)+PX
200 Y3=45*SIN(A)+PY
210 X4=150*SIN(A+PI/2)*3+PX
220 Y4=45*COS(A+PI/2)*3+PY
230 LINE(X2, Y2)-(X1, Y1)
240 LINE-(X3, Y3)

```

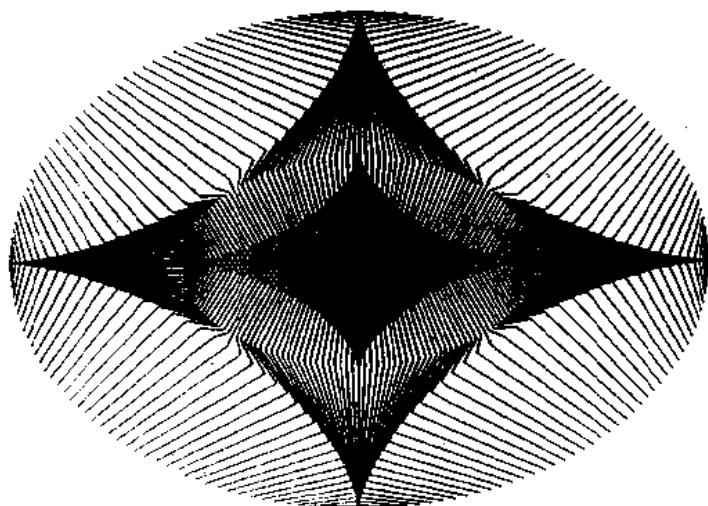


图 3.27

250 LINE-(X4, Y4)

260 NEXT A

270 END

程序运行结果如图 3.27.

例 3-25 具有立体感的图案

100 'pc3-25

110 SCREEN 2: CLS

120 PI=3.14159

130 FOR A=0 TO 2*PI STEP PI/120

140 X1=320+30 * COS(A)

150 Y1=100+15 * SIN(A)

160 R=60 * (1+1 / 6 * SIN(3 * A))

170 S=100 * (1+1 / 6 * SIN(4 * A))

180 T=140 * (1+1 / 6 * SIN(5 * A))

190 U=180 * (1+1 / 8 * SIN(6 * A))

200 X2=R * COS(A+PI / 20)+320

210 Y2=R * SIN(A+PI / 20) / 2+100

220 X3=S * COS(A)+320

230 Y3=S * SIN(A) / 2+100

240 X4=T * COS(A+PI / 20)+320

250 Y4=T * SIN(A+PI / 20) / 2+100

260 X5=1.5 * U * COS(A)+320

270 Y5=U * SIN(A) / 2+100

280 LINE(X1, Y1)-(X2, Y2)

290 LINE-(X3, Y3)

300 LINE-(X4, Y4)

310 LINE-(X5, Y5)

320 NEXT A

330 END

程序运行结果如图 3.28，这是一个十分奇妙的具有立体感的图案。

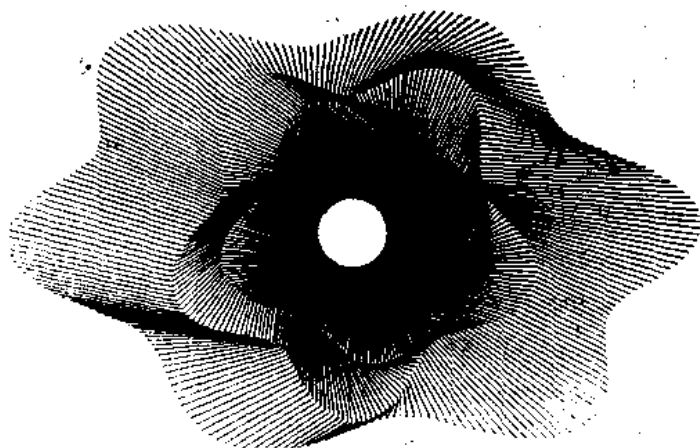


图 3.28

例 3-26 四瓣花型图案

100 'pc3-26

110 SCREEN 2: CLS

120 PI=3.14159 D=100

130 FOR A=0 TO 2*PI STEP PI/360

140 E=D*(1+SIN(4*A))

150 X1=320+E*COS(A)

160 X2=320+E*COS(A+PI/5)

170 Y1=100+E*SIN(A)/2

180 Y2=100+E*SIN(A+PI/5)/2

190 LINE(X1, Y1)-(X2, Y2)

200 NEXT A

210 END

这是一个画 4 瓣花型图案的例子。程序中 140 行的 $\text{SIN}(4 * A)$ 的 4 可取 3~10 范围内的整数值，依次画出 3~8 瓣的图案。程序运行结果如图 3.29

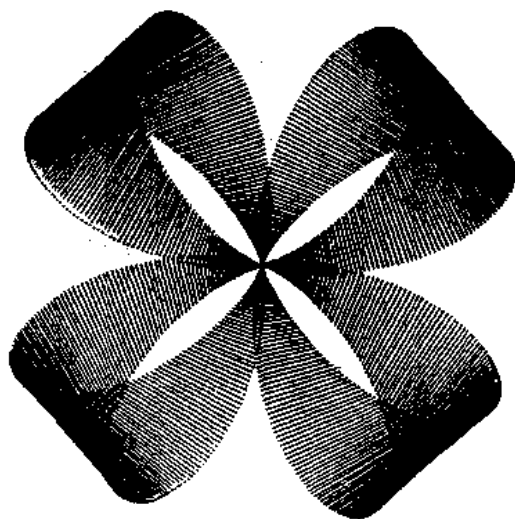


图 3.29

例 3-27 顶端有折射皱的花型

100 'pc3-27

110 SCREEN 2: CLS

120 $\text{PI}=3.14159$: $D=100$

130 FOR $A=0$ TO $2 * \text{PI}$ STEP $\text{PI} / 360$

140 $E=D * (1 + 1 / 4 * \text{SIN}(12 * A))$

150 $F=E * (1 + \text{SIN}(4 * A))$

160 $X1=320+F * \text{COS} * A$

170 $X2=320+F * \text{COS}(A+\text{PI} / 5)$

```

180 Y1=100+F*SIN(A)/2
190 Y2=100+F*SIN(A+PI/5)/2
200 LINE (X1, Y1)-(X2, Y2)
220 END

```

程序执行结果如图 9.30。这是一个瓣顶有折皱的四瓣花型图案。140 行语句中 $\text{SIN}(12 * A)$ 的数字 12，意义为 4×3 。4 表示瓣数，3 是变形系数。

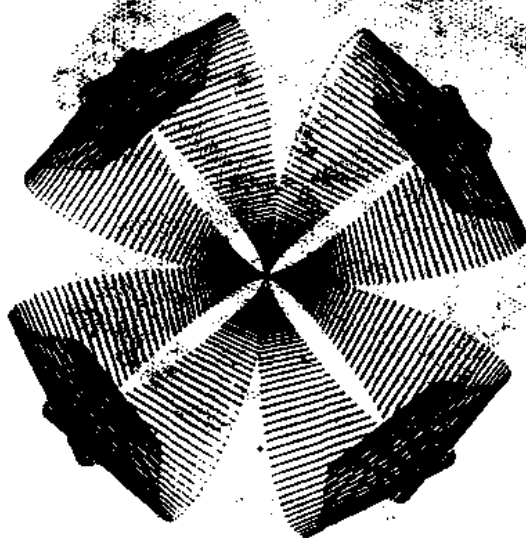


图 3.30

例 3-28 变形四瓣花型图案 1

```

100 'pc3-28
110 SCREEN 2: CLS
120 PI=3.14159: D=80
130 FOR A=0 TO 2*PI STEP PI/360
140 E=D*(1+1/4*COS(24*A))
150 F=E*(1+SIN(4*A))

```

```

160 X1 = 320 + F * COS(A)
170 X2 = 320 + F * COS(A + PI / 5)
180 Y1 = 100 - F * SIN(A) / 2
190 Y2 = 100 - F * SIN(A + PI / 5) / 2
200 LINE (X1, Y1)-(X2, Y2)
210 NEXT A
220 END

```

程序运行结果如图 3.31，140 行语句中的数 24 表示 4 瓣 $\times 6$ ，6 是变形系数。如果将变形系数改为 5 ($20 * A$) 或 4 ($16 * A$)，可得到不同的图案。

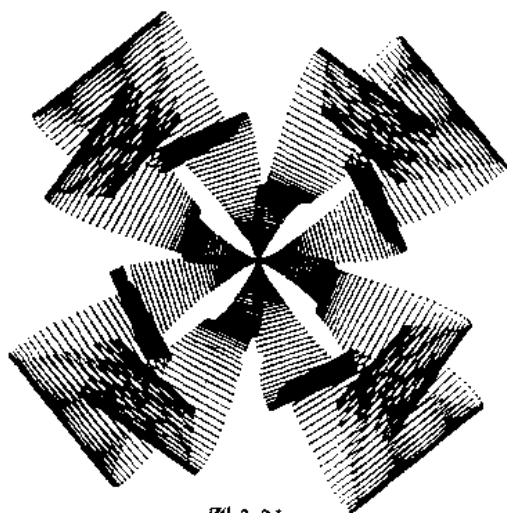


图 3.31

例 3-29 变形四瓣花型图案 2

```

100 'pc3-29
110 SCREEN 2: CLS: KEY OFF
120 PI = 3.14159: D = 80

```

```

130 FOR A=0 TO 2*PI STEP PI/360
140 E=D*(1+1/4*COS(12*A))
150 F=E*(1+SIN(4*A))
160 X1=320-F*COS(A)
170 X2=320+F*COS(A+PI/5)
180 Y1=100-F*SIN(A)/2
190 Y2=100-F*SIN(A+PI/5)/2
200 LINE (X1, Y1)-(X2, Y2)
210 NEXT A
220 END

```

本程序与前例程序相比，仅 140 行中的三角函数不同。程序运行结果如图 3.32。改变 140 行中 $\cos(12 * A)$ 的 $12 * A$ 为 $(16 * A)$ 或 $(20 * A)$ ，可生成不同的图案。

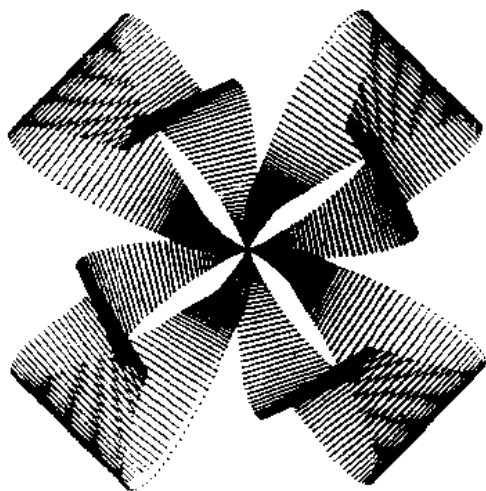


图 3.32

例 3-30 变形四瓣花型图案 3


```

100 'pc3-30
110 SCREEN 2: CLS
120 PI=3.14159: D=100
130 FOR A=0 TO 2*PI STEP PI/30
140 E=D*(1+1/3*COS(12*A))
150 F=E*(1+3/7*SIN(4*A))
160 X1=320+F*COS(A)
170 X2=320+F*COS(A+PI/3)
180 Y1=100-F*SIN(A)/2
190 Y2=100-F*SIN(A+PI/3)/2
200 LINE (X1, Y1)-(X2, Y2)
210 NEXT A
220 END

```

运行结果如图 3.33。

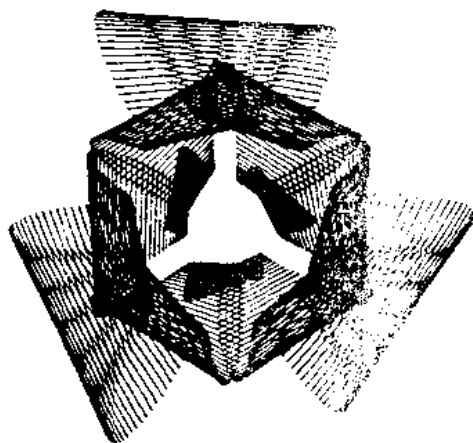


图 3.33

例 3-31 具有长短瓣的花型图案

```
100 'pc3-31
110 SCREEN 2: CLS
120 PI=3.14159: D=80
130 FOR A=0 TO 2*PI STEP PI/300
140 E=D*(1+1/4*COS(3*A))
150 F=E*(1+SIN(6*A))
160 X1=320+F*COS(A)
170 X2=320+F*COS(A+PI/6)
180 Y1=100-F*SIN(A)/2
190 Y2=100-F*SIN(A)/2
200 LINE (X1, Y1)-(X2, Y2)
210 NEXT A
220 END
```

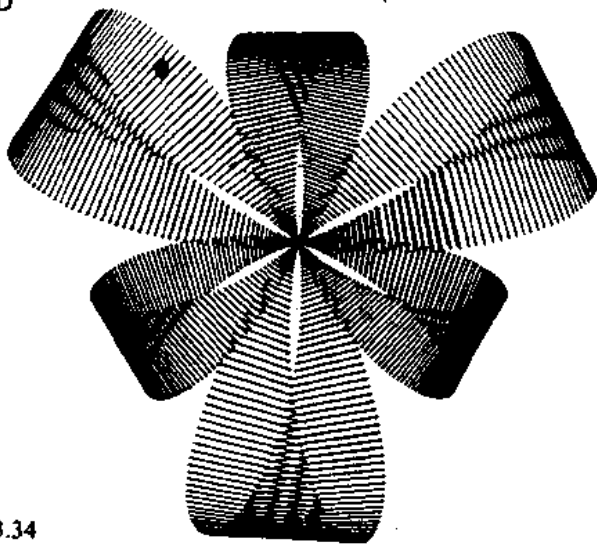


图 3.34

这是一个绘有长瓣、短瓣花型图案的程序，程序运行结果如图 3.34。

例 3-32 八瓣花型图案

```
100 'pc3-32
110 SCREEN 2: CLS
120 PI=3.14159: D=80
130 FOR A=0 TO 2*PI STEP 2*PI/720
140 E=D*(1+1/4*COS(4*A))
150 F=E*(1+SIN(8*A))
160 X1=320+F*COS(A)
170 X2=320+F*COS(A+PI/8)
180 Y1=200-F*SIN(A+PI/8)
190 Y2=200-F*SIN(A+PI/8)
200 LINE(X1, Y1/2)-(X2, Y2/2)
```

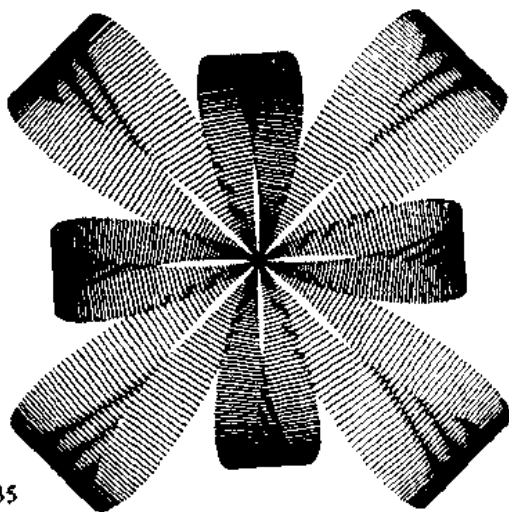


图 3.35

210 NEXT A

220 END

这个程序画一个带长短瓣的 8 瓣花型图案程序运行结果如图 3.35。

例 3-33 莫尔花纹图案

100 'pc3-33

110 SCREEN 2: CLS

120 PI=3.14159: D=120

130 FOR A=0 TO 4*PI STEP PI/120

140 E=D*(1+1/2*SIN(2.5*A))

150 X1=320+E*COS(A)

160 X2=320+E*COS(A+PI/6)

170 Y1=100-E*SIN(A)/2

180 Y2=100-E*SIN(A+PI/4)/2

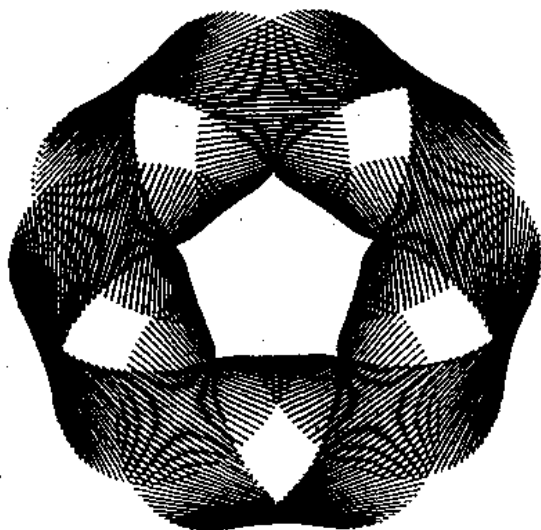


图 3.36

```

190 LINE (X1, Y1)-(X2, Y2)
200 NEXT A
210 END

```

程序运行结果如图 3.36,

例 3-34 曲线组成的图案

利用函数的曲线, 例如三角函数、二阶或三阶函数曲线的平移、旋转等, 可以构成许多种图案。当取函数为正弦函数时, 程序如 pc3-34。

```

100 'pc3-34
110 SCREEN 2: CLS
120 PI=3.14159: D=120
130 FOR PY=20 TO 380 STEP 5
140 B=4*PI*(PY-20)/360

```

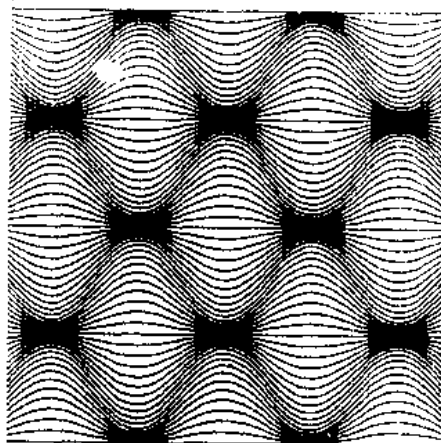


图 3.37

```

150 FOR A=0 TO 5*PI STEP PI/16
160 X=360/(5*PI)*A+140
170 Y=PY+25*SIN(B)*SIN(A)
180 IF A=0 THEN PSET (X, Y/2)

```

```

190 LINE-(X, Y / 2)
200 NEXT A
210 NEXT PY
220 END

```

程序运行结果如图 3.37.

例 3-35 具有立体感的图案

```

100 'pc3-35
110 SCREEN 2: CLS
120 PI=3.14159
130 FOR PY=20 TO 380 STEP 5
140 AL=2 * PI * (PY-20 / 360
145 L=PI * SIN(AL)
150 FOR A=0 TO 6 * PI STEP PI / 8
160 X=440 / (6 * PI) * A+140

```

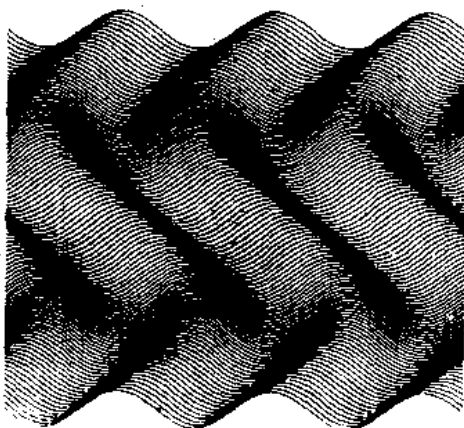


图 3.38

```

170 Y=PY+18 * SIN(A+L)
180 IF A=0 THEN PSET (X, Y / 2)
190 LINE-(X, Y / 2)
200 NEXT A
210 NEXT PY
220 END

```

程序运行结果如图 3.38。波峰、波谷的位置按正弦曲线规律变化、同时，波峰亦呈波动形状。

例 3-36 意形图案 1

```

100 'pc3-36
110 SCREEN 2: CLS
120 PI=3.14159
130 FOR PY=20 TO 190 STEP 3
140 AL=2 * PI * (PY-20) / 360
150 L=PI * SIN(AL)
160 FOR A=0 TO 6 * PI STEP PI / 7
170 X=600 / (6 * PI) * A+20
180 Y=PY+30 * SIN(AL) * SIN(A+SIN(A) * PI~L)
190 IF A=0 THEN PSET (X, Y)
200 LINE-(X, Y)
210 NEXT A
220 NEXT PY
230 END

```

程序运行结果如图 3.39。这是一个将卵放在布上的图案。程序中仅 180 行语句与前例不同。变化振幅和相位角，可以生成多种不同的图案。

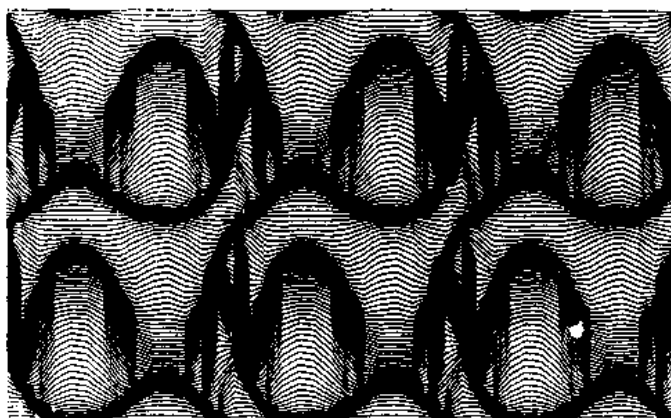


图 3.39

例 3-37 意形图案 2

```

100 'pc3-37
110 SCREEN 2: CLS
120 PI=3.14159
130 FOR PY=20 TO 380 STEP 3
140 AL=2*PI*(PY-20)/360
150 L=PI*SIN(AL)
160 FOR A=0 TO 12*PI STEP PI/30
170 X=600/(12*PI)*A+20
180 Y=PY-10*SIN(A+SIN(A)*PI-L)
190 IF A=0 THEN PSET (X, Y/2)
200 LINE-(X, Y/2)
210 NEXT A
220 NEXT PY
230 END

```


程序运行结果为图 3.40, 为前例的变形图案。图案较清晰, 但程序运行时间较长。

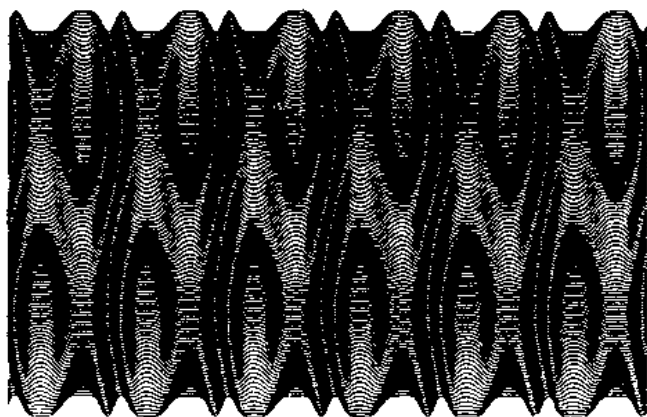


图 3.40

例 3-38 异形图案 3

```
100 'pc3-38
110 SCREEN 1,0 :CLS
120 PI=3.14159 :CO=1
130 FOR RY=20 TO 380 STEP 3
140 IF PY>70 THEN YY=10 :CO=2
150 IF PY>130 THEN YY=20 :CO=3
160 AL=(PY-20)*2*PI/360
170 L=PI*SIN(AL)
180 FOR A=0 TO 12*PI STEP PI/30
190 X=639/(12*PI)*A
200 Y=YY+10*SIN(A+SIN(A)*PI-L)+PY
210 IF A=0 THEN PSET(X/2,Y/2)
220 LINE-(X/2,Y/2),CO
230 NEXT A
240 NEXT PY
250 END
```

程序运行结果如图 3.41。

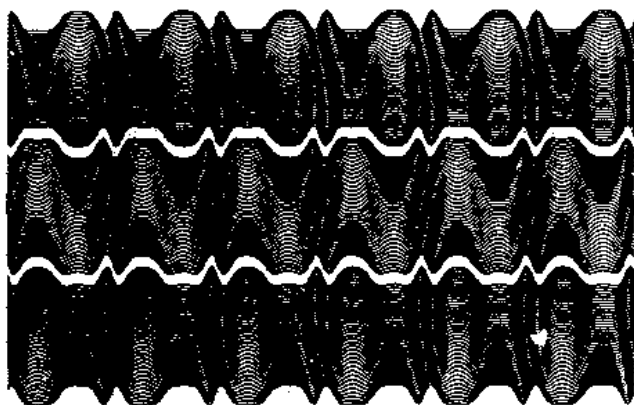


图 3.41

例 3-39 沙丘风纹图案

```
100 'pc3-39
100 'pc3-39
110 SCREEN 2 :CLS
120 PI=3.14159
130 FOR RY=20 TO 380 STEP 3
160 AL=(PY-20)*4*PI/360
170 L=PI*COS(AL)
180 FOR A=0 TO 6*PI STEP PI/7
190 X=600/(6*PI)*A+20
200 Y=10*SIN(A+COS(A)*PI-L)*COS(AL)+PY
210 IF A=0 THEN PSET(X,Y/2)
220 LINE STEP(0,0)-(X,Y/2)
230 NEXT A
240 NEXT PY
250 END
```

沙丘风纹图案程序。当 COS 函数的起始角度为 0 时，图案边缘凸凹不平。如果在 160 行的最后追加 $-\pi/2$ ，图案边缘会变得平整。程序运行结果如图 3.42。

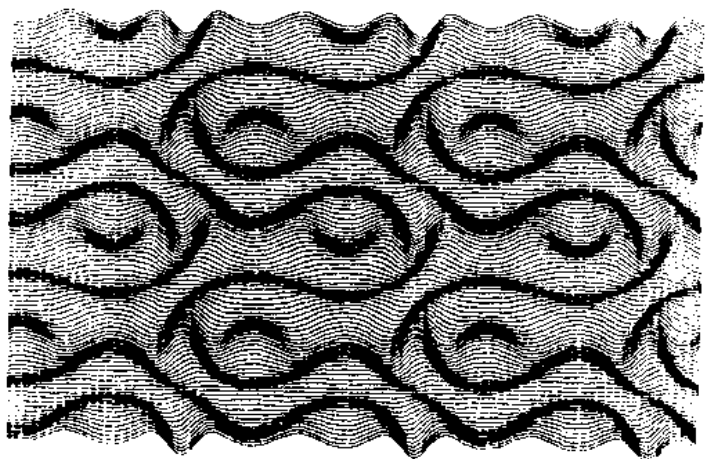


图 3.42

例 3-40 曲线图案

```

100 'pc3-40
110 SCREEN 2 :CLS
120 PI=3.14159 :NN=50 :AM=4
130 DIM Z(NN+1)
140 FOR M=1 TO NN+1
150 Z(M)=360/NN*(M-1)
160 NEXT M
170 FOR M=1 TO NN+1
180 FOR N=1 TO NN+1
190 L=AM*SIN(2*PI/360*Z(N))*SIN(PI/360*Z(M))
200 PX=L+Z(M)+140
210 PY=(L+Z(N)+20)/2
220 IF N=1 THEN PSET(PX,PY)
230 LINE-(PX,PY)
240 NEXT N,M

```

程序运行结果如图 3.43.

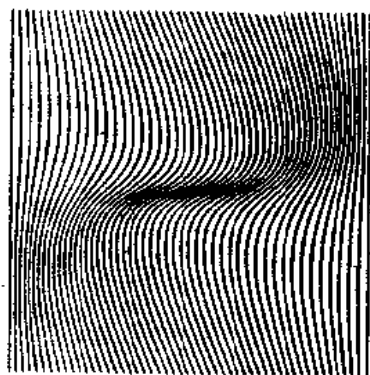


图 3.43

例 3-41 李沙育图形图案 1

```
100  'pc3-41
110  SCREEN 2 :CLS
120  PI=3.14159:N=1
130  FOR PX=200 TO 440 STEP 120
150  FOR PY=80 TO 320 STEP 120
160  IF N=8 THEN N=2 ELSE N=N+1
170  FOR A=0 TO 2*PI STEP PI/(20*N)
180  X=PX+55*SIN(N*A)
190  Y=PY-55*SIN((N+1)*A)
200  IF A=0 THEN PSET(X,Y/2):BX=X:BY=Y/2
210  LINE-(X,Y/2)
220  NEXT A:LINE-(BX,BY)
230  NEXT PX,PY
240  END
```

根据式 $X = D \sin(N * \alpha)$

$$Y = S \sin (M * \alpha) \quad (0 \leq \alpha < 2\pi)$$

$$M = N+1$$

画出的李沙育曲线。程序运行结果如图 3-44。

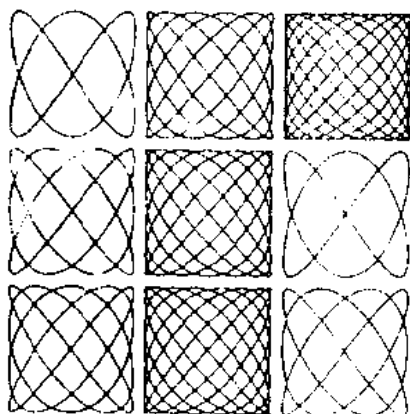


图 3-44

例 3-42 李沙育图形图案 2

```

100  'pc3-42
110  SCREEN 2 :CLS
120  PI = 3.14159
130  FOR PX = 170 TO 470 STEP 60
140    I = I + 1
150    FOR PY = 50 TO 350 STEP 60
160      IF N = 6 THEN N = 1 ELSE N = N + 1
170      FOR A = 0 TO 2 * PI STEP PI / (20 * N)
180        X = PX + 28 * SIN(N * A)
190        Y = PY + 28 * SIN(I * A)
200        IF A = 0 THEN PSET (X, Y / 2) : BX = X : BY = Y / 2
210        LINE-(X, Y / 2)
220      NEXT A : LINE-(BX, BY)
230    NEXT PY, PX
240  END

```

程序中按式 $X = R \cdot \sin(N * \alpha)$

$Y = R \cdot \sin(I * \alpha) \quad (0 < \alpha < 2\pi)$

并且依从左到右 $I = 1, 2, \dots, 6$, 从上到下 $N = 1, 2, \dots, 6$ 的规律配置图形, 如图 3.45。左上至右下对角线上的图形 ($I = N$) 似乎是一条线, 但实际是所围面积为 0 的闭合曲线。画图时左上闭合曲线画一次, 依次至右下角闭合曲线增加到重复画六次。

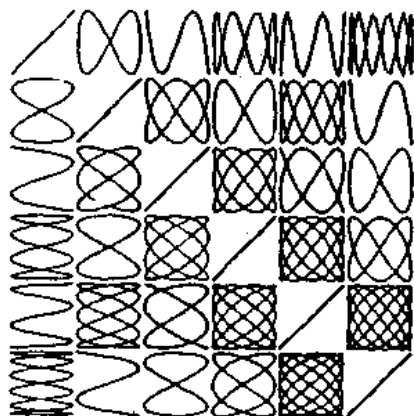


图 3.45

例 3-43 李沙育图形图案 3

```

100  'pc3-43
110  SCREEN 2 :CLS
120  PI=3.14159
130  FOR PX=170 TO 470 STEP 60
140  I=I+1
150  FOR PY=50 TO 350 STEP 60
160  IF N=6 THEN N=1 ELSE N=N+1
170  FOR A=0 TO 2*PI STEP PI/(20*N)
180  X=PX+28*SIN(N*A)
190  Y=PY-28*COS(I*A)
200  IF A=0 THEN PSET(X,Y/2):BX=X:BY=Y/2
210  LINE-(X,Y/2)

```

```

220   NEXT A:LINE-(BX, BY)
230   NEXT PY,PX
240   END

```

程序运行结果如图 3.46。与前例程序差别为 180 行和 190 行中分别使用 SIN 和 COS 函数图形配置与前例相同。

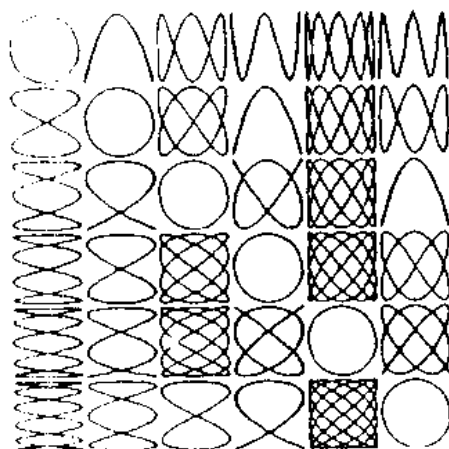


图 3.46

例 3-44 螺线图

```

100   'pc3-44
110   SCREEN 2 :CLS
120   PI = 3.14159
130   FOR A = 0 TO 6 * PI STEP PI / 12
140     R = R + 1
150     PX = 300 + (R + 20) * COS(A)
160     PY = 90 + (R / 2 + 10) * SIN(A)
170     BT = BT + PI / 24
180     FOR AL = 0 TO 2 * PI STEP PI / 15
190       X = PX + R / 6 * COS(AL + A)
200       Y = PY - R / 12 * COS(AL + BT)

```

```

210  IF A=0 THEN PSET (X,Y): BX=X:BY=Y
220  LINE-(X,Y)
230  NEXT AL
240  NEXT A
250  END

```

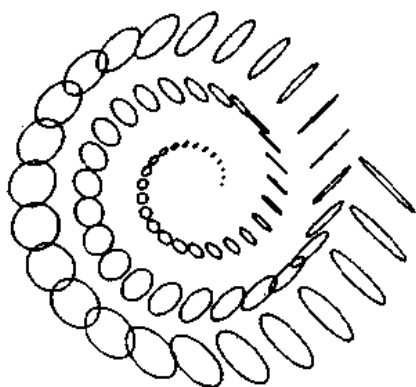


图 3.47

这是一个在螺线上依次描画变化的李沙如曲线的程序。运行结果如图 3.47 所示。程序中 150、160 行确定李沙如曲线的位置，由 140 行使 R 每次加 2 形成螺线状图案。李沙如曲线也因 R 值每次增大 $R/6$ 逐渐变大。(190 行语句)

例 3-45 蔷薇花

```

100  'pc3-45
110  SCREEN 2 :CLS
120  PI=3.14159:B=50:M=1
130  FOR PX=200 TO 440 STEP 120
140  FOR PY=40 TO 160 STEP 60
150  M=M+1:IF M MOD 2=0 THEN K=2 ELSE
      K=1

```



```

160  FOR A=0 TO K * PI STEP PI / (30 * M)
170  X=PX+B * SIN(M * A) * COS(A)
180  Y=PY+B * SIN(M * A) * SIN(A) / 2
190  IF A=0 THEN PSET (X,Y) : BX=X:BY=Y
200  LINE-(X,Y)
220  NEXT PY,PY
250  END

```

程序中由 M 值决定花的瓣数。当 M 为奇数时，花瓣数为 M ；当 M 值取偶数时，花瓣数为 $2M$ 。程序运行结果如图 3.48。

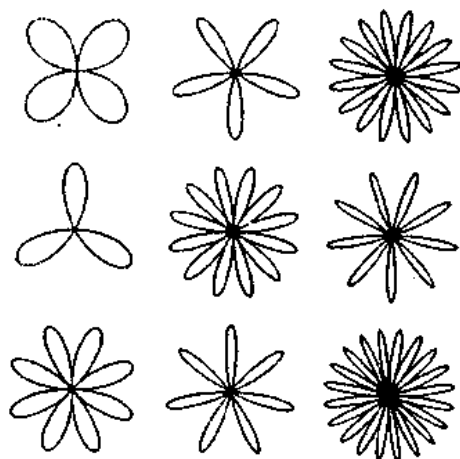


图 3.48

例 3-46 变形蔷薇花

```

100  'pc3-46
110  SCREEN 2 :CLS
120  PI=3.14159:B=28:N=1

```

```

130  FOR PX=170 TO 470 STEP 60
140  FOR PY=25 TO 175 STEP 30
150  P=P+1:IF N MOD 2=0 THEN K=2 ELSE K=1
160  FOR A=0 TO K*PI STEP PI/(15*M)
170  X=PX+(B/5*SIN(N*P*A)+B*SIN(N*
    A))*COS(A)
180  Y=PY+(B/5*SIN(N*P*A)+B*SIN(N*
    A))*SIN(A)/2
190  IF A=0 THEN PSET (X,Y):BX=X:BY=Y
200  LINE-(X,Y)
210  NEXT A:LINE-(BX,BY)
220  NEXT PY,PY
250  END

```

程序中 170 行、180 行中的 $B/5 * \sin(N * P * A)$ 为变形因子。 $B/5$ 中的 5 是变形量， P 参数值改变花形。运行结果如图 3.49。

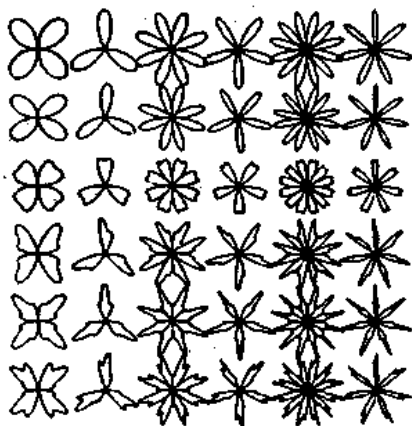


图 3.49

例 3-47 旋转直线图案

```
100 'pc3-47
110 SCREEN 1,0:CLS
120 PAI=3.14159
130 RMIN=10:RMAX=100:RS=40
140 XMID=160
150
160 FOR R=RMIN TO RMAX STEP RS
170 CLS
180 YL=30:YH=0
190 FOR T=0 TO 8*PAI STEP PAI/20
200 YL=YL+1:YH=YH+1
210 X=R*COS(T)+XMID
220 LINE (XMID,YL)-(X,YH)
230 PSET (X,YH)
240 NEXT T
250 FOR T=1 TO 1000:NEXT T
260 NEXT R
270 END
```

程序运行结果如图 3.50。

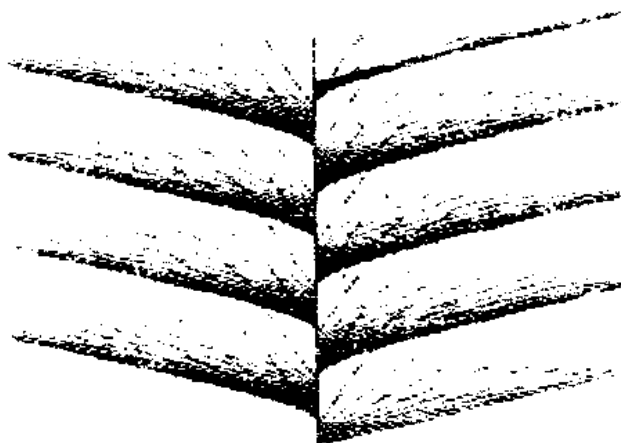


图 3.50

例 3-48 落叶

```
100 'pc3-48
110 SCREEN 2:CLS
120 XMAX=440:YMAX=180
130 XMIN=200:YMIN=20
140 XW=(XMAX-XMIN):YW=(YMAX-YMIN)
150 '
160 CLS
170 FOR I=1 TO 30
180   X=INT(RND(I)*XW)+XMIN
190   Y=INT(RND(I)*YW)+YMIN
200   GOSUB 250
220 NEXT I
230 END
240
250   LINE(X+40,Y)-(X+32,Y+4)
260   LINE-(X+20,Y+3)
270   LINE-(X+22,Y+4)
280   LINE-(X+12,Y+5)
290   LINE-(X+16,Y+6)
300   LINE-(X+8,Y+9)
310   LINE-(X+8,Y+12)
320   LINE-(X+6,Y+13)
330   LINE-(X+16,Y+12)
340   LINE-(X+18,Y+13)
350   LINE-(X+20,Y+10)
360   LINE-(X+24,Y+11)
370   LINE-(X+26,Y+8)
380   LINE-(X+30,Y+9)
390   LINE-(X+32,Y+4)
400   LINE-(X+29,Y+5)-(X+20,Y+7)
410   LINE-(X+7,Y+13)
420   RETURN
```

运行结果如图 3.51

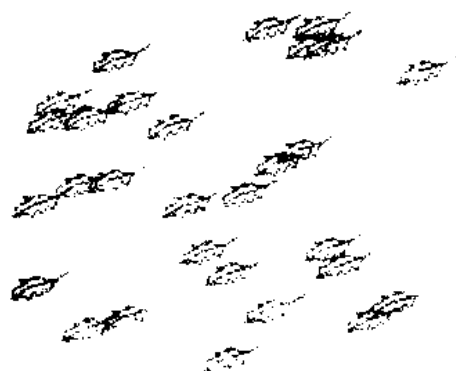


图 3.51

例 3-49 二阶函数曲线图案

将二阶函数曲线平移，做成图形单元，旋转图形单元，构成一幅美妙的图案。

```

100  'pc3-49
110  SCREEN 2 :CLS
120  PI=3.14159
130  INPUT "n"; N
140  TH=PI*(.5-1/N):A=TAN(TH)
150  INPUT "b,c,l"; B,C,L
160  IF B<L*A THEN PRINT "can't":GOTO 150
170  M=A^2/(4*(B-L*A))
180  FOR AA=0 TO 2*PI*(N-1)/N+.2 STEP
      2*PI/N
190  FOR LL=-L TO L+4.9999 STEP 5
200  MX1=SQR((C-B)/M+LL:M*2
      =-SQR((C-B)/M)+LL

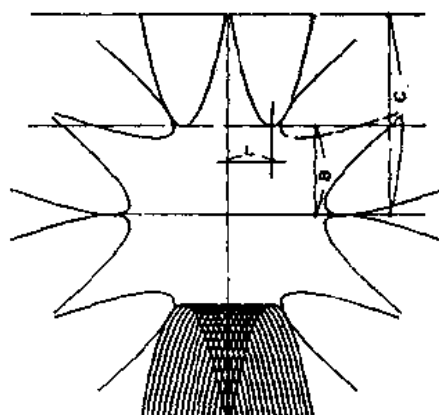
```

```

210  FOR X=MX2 TO M×1 STEP (M×1-M×
      2)/20
220  Y=M*(X-LL)2+B:GOSUB 310
230  IF X=MX2 THEN PSET (PX, PY):BX
      =X+(MX1-MX2):BY=Y
240  LINE-(PX, PY)
250  NEXT X:X=BX:Y=BY
260  GOSUB 310:LINE-(PX, PY)
270  NEXT LL
280  NEXT AA
290  END
310  PX=X * COS(AA)-Y * SIN(AA)+320
320  PY=X * SIN(AA)+Y * COS(AA)+200:PY=PY
      /2
330  RETURN

```

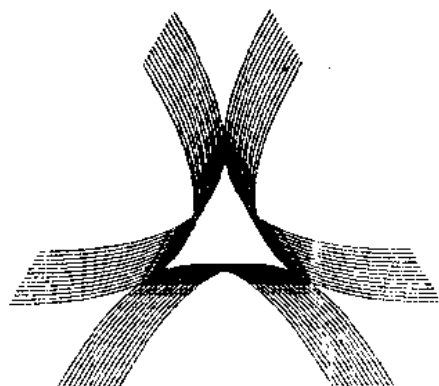
程序中 N 为单元数，参数 B、C、L 的意义如图 3.52 所示。



N=6

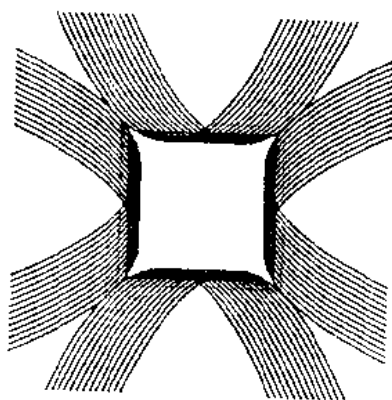
图 3.52

程序运行结果例：图 3-53~ 3.55



$N=3, B=30, C=150, L=30$

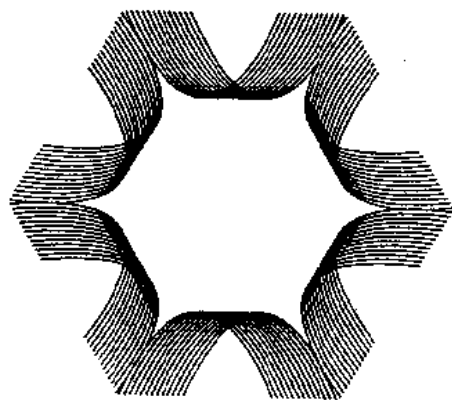
图 3.53



$N=4, B=50,$
 $C=150, L=30$

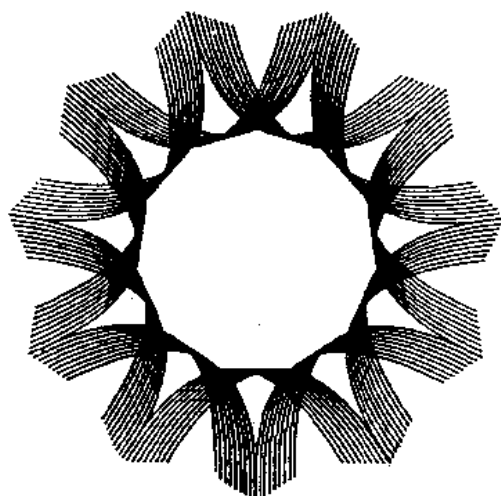
图 3.54

如果将程序中 180 行循环终值取



$N=6$, $B=100$, $C=180$, $L=30$

图 3.55



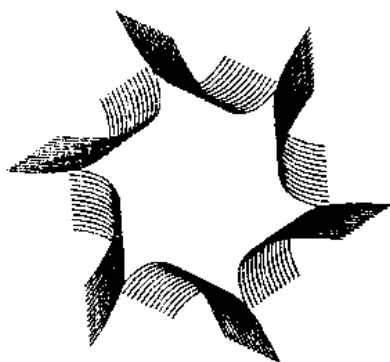
$N=5.5$, $B=100$, $C=180$, $L=20$

图 3.56

$$4 * \text{PI} * (N - 0.5)$$

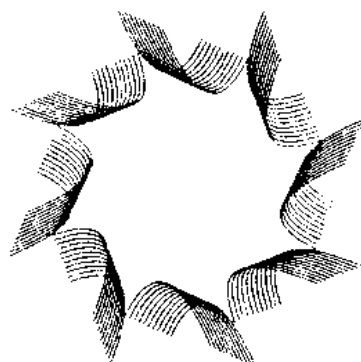
取 $N = 3.5$ 、 4.5 或 5.5 ，则程序运行结果如图 3.56 所示。

若将程序中 210 行的 STEP 固定为 5，220 行加上 $X/2$ ，删除 260 行，程序运行结果如图 3.57 和图 3.58



$N = 6$, $B = 100$,
 $C = 160$, $L = 30$

图 3.57



$N = 8$, $B = 120$,
 $C = 180$, $L = 25$

图 3.58

例 3-50 变形二阶函数曲线图案

```

100 'pc3-50
110 SCREEN 2 :CLS
120 PI = 3.14159
130 INPUT "n"; N
140 TH = PI * (.5 - 1 / N) : A = TAN(TH)
150 INPUT "b,c,l"; B, C, L
160 IF B < L * A THEN PRINT "can't": GOTO 150
170 M = A^2 / (4 * (B - L * A))
180 MX1 = SQR((C - B) / M) + L : MX2 = -SQR((C - B)
    / M + L
190 FOR AA = 0 TO 2 * PI * (N - 1) / N + .2 STEP
    2 * PI / N

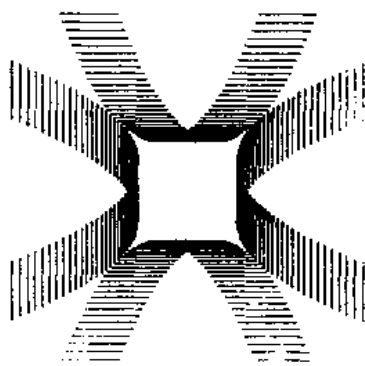
```

```

200  FOR X=MX2 TO MX1 STEP (MX1-MX2)/60
210  X2=X-2*L
220  Y=M*(X-L)2+B:GOSUB 270
230  LINE (PX, PY)-(PX2, PY2)
240  NEXT X, AA
250  END
270  PX=X * COS(AA)-Y * SIN(AA)+320
280  PX2=X2 * COS(AA)-Y * SIN(AA)+320
290  PY=X * SIN(AA)+Y * COS(AA)+200:PY=
    PY/2
300  PY2=X2 * SIN(AA)+Y * COS(AA)+200:PY2=
    Y2/2
330  RETURN

```

程序运行结果如图 3.59。



$N=4$, $B=50$, $C=180$, $L=30$

图 3.59

例 3-51 奇异二阶函数曲线图案

```

100  'pc3-51
110  SCREEN 2 :CLS

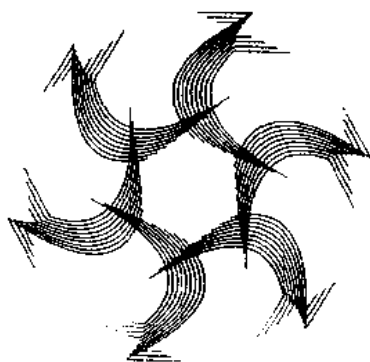
```

```

120  PI = 3.14159
130  INPUT "n"; N
140  TH = PI * (.5-1 / N) : A = TAN(TH)
150  INPUT "b,c,l"; B, C, L
160  IF B < L * A THEN PRINT "can't":GOTO 150
170  M = A^2 / (4 * (B-L * A))
180  FOR AA = 0 TO 2 * PI * (N-1) / N+.2 STEP
      2 * PI / N
190  FOR LL = -L TO L+.9999 STEP 5
200  MX1 = SQR((C-B) / M)+LL:MX2 = -SQR((C-B)
      / M+LL
210  FOR X = MX2 TO MX1 STEP (MX1-MX2) / 20
220  Y = M * (X)^2:GOSUB 300
230  IF X = MX2 THEN PSET (EX,EY):BX
      = X+(MX1-MX2):BY = Y
240  LINE-(EX,EY)
250  NEXT X:X = BX:Y = BY :GOSUB 300
260  LINE-(EX,EY):NEXT LL
280  NEXT AA
290  END
300  PX = X * COS(PI / 2)-Y * SIN(PI / 2)+LL
310  PY = X * SIN(PI / 2)+Y * COS(PI / 2)+B
320  GOSUB 340
330  RETURN
340  EX = PX * COS(AA)-PY * SIN(AA)+320
350  EY = PX * SIN(AA)+PY * COS(AA)+200:EY = E
      Y / 2
360  RETURN

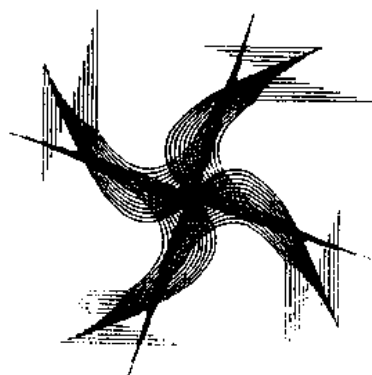
```

程序运行结果如图 3.60 及图 3.61(参数见图)



$N=6, B=100, C=150, L=20$

图 3.60



$N=4, B=50, C=150, L=30$

图 3.61

例 3-52 三阶函数曲线组成的图案

```

100 'pc3-52
110 SCREEN 2 :CLS
120 X0=120:X1=180:Y1=180
130 A=Y1/(X1*(X1-X0)*(X1+X0))
140 FOR XX=-50 TO 50 STEP 5
160 FOR X=-180 TO 180 STEP 2
170 Y=A*X*(X-X0)*(X+X0)
180 PX=320+X+XX:PY=100-Y/2
190 IF X=-180 THEN PSET(PX,PY*.75)
200 LINE-(PX,PY*.75)
210 NEXT X
220 NEXT XX
240 END

```

程序运行结果如图 3.62。这是将 3 阶函数曲线等间隔

(步长 5) 进行平移变换而构成的图案。

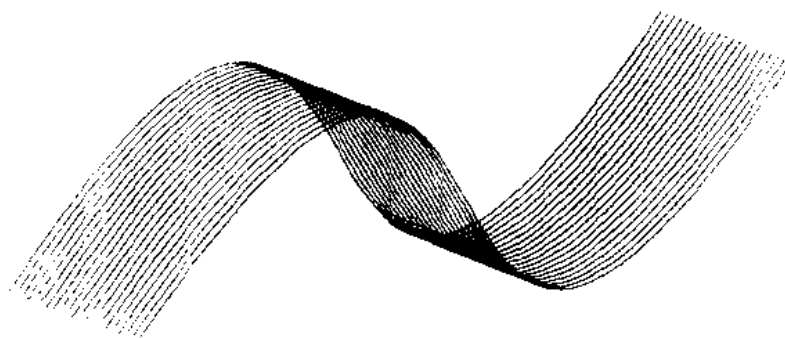


图 3.62

例 3-53 三阶函数曲线旋平移图案

```
100 'pc3-53
110 SCREEN 2 :CLS
120 X0=120:X1=180:Y1=120
130 A=Y1/(X1*(X1-X0)*(X1+X0))
140 FOR YY=-75 TO 75 STEP 75
150 FOR XX=-50 TO 50 STEP 5
160 FOR X=-180 TO 180 STEP 2
170 Y=A*X*(X-X0)*(X+X0)
180 PX=320+X+XX:PY=100-Y/2+YY/2
190 IF X=-180 THEN PSET (PX,PY)
200 LINE-(PX,PY)
210 NEXT X
```

将前例的图案在 Y 轴方向进行平移。程序运行结果如图 3.63。

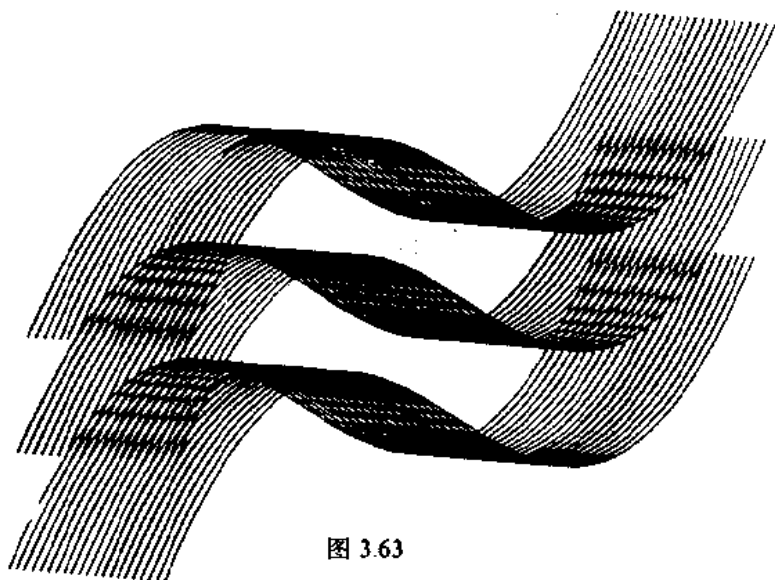


图 3.63

例 3-54 五阶函数曲线图案

```

100  'pc3-54
110  SCREEN 2 :CLS
120  X0=160:X1=180:X2=80:Y1=180
130  A=Y1/(X1*(X1-X0)*(X1+X0)*(X1-X2)*(
      X1+X2))
140  FOR YY=-75 TO 75 STEP 75
150  IF YY<10 AND YY>-10 THEN 230
160  FOR X=-180 TO 180 STEP 2
180  Y=A*X*(X-X0)*(X+X0)*(X-X2)*(X+X2)
190  PX=X+320:PY=100-Y/2+YY/2
200  IF X=-180 THEN PSET (PX,PY)
210  LINE--(PX,PY)
220  NEXT X
230  NEXT YY
240  END

```

由 5 阶函数曲线进行上下平移。运行结果如图 3.64

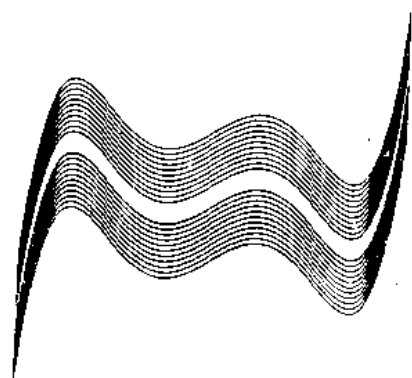


图 3.64

例 3-55 组合函数曲线图案

组合函数曲线图案，如图 3.65。

```
100 'pc3-55
110 SCREEN 2 :CLS
130 PI= 3.14159
140 FOR P=-1 TO 1 STEP 2
150 FOR N=1 TO 2
160 FOR FY=-45 TO 45 STEP 5
170 FOR A=-180 TO 180 STEP 6
180 X1=P * A
190 Y1=135 / 180 * A+FY
200 Y2=30 * SIN(A * 4 * PI / 360)
210 PY=Y1+Y2
220 IF N=2 THEN SWAP X1,PY
230 X=X1+320 :Y=PY+200
240 IF A=-180 THEN PSET(X,Y / 2)
250 LINE-(X,Y / 2)
```

260 NEXT A,FY,N,P

270 END

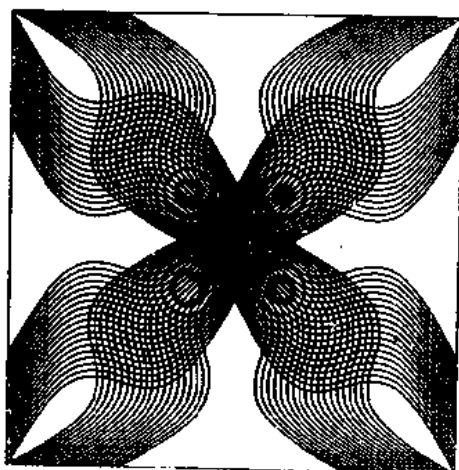


图 3.65

例 3-56 组合函数曲线图案 2

程序运行结果如图 3.66

```
100 'pc3-56
110 SCREEN 2 :CLS
130 PI=3.14159
140 FOR P=-1 TO 1 STEP 2
150 FOR N=1 TO 2
160 FOR FY=0 TO 360 STEP 10
170 FOR A=-180 TO 180 STEP 12
180 X1=P*A
190 Y1=FY/2+180/360*(A-180)
200 Y2=30*SIN(A*4*PI/360)
210 PY=Y1+Y2
220 IF N=2 THEN SWAP X1,PY
```

—186—


```

230  X=X1+320:Y=PY+200
240  IF A=-180 THEN PSET(X,Y/2)
250  LINE-(X,Y/2)
260  NEXT A,FY,N,P
270  END

```

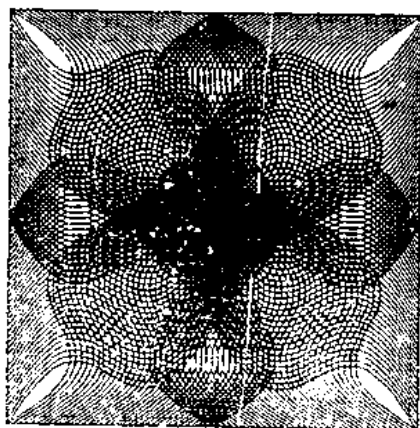


图 3.66

例 3-57 印花图案及花边图案程序包

可以描绘成千上万种印花图案及花边图案的程序包如 PC3-57。程序运行时，由人一机对话方式依次输入 8 个参数，参数的组合不同，生成的图案也不同。各参数的取值范围如下：

1	UN	
	屏幕显示时:	1~6 (整数)
	绘图机输出:	1~12 (整数)
2	PO	1~4
3	TP	1~5

*****	123
	234
	543
	45
	23
	43
*****	567
*****	897
*****	876
*****	546
*****	345
	234
-----+-----+-----+-----+	
.4 358.8 538.2 717.6 897	

8

```

490 INPUT "sf=0.2-1.0"; SF
500 CLS: PRINT
510 FOR BT=BE TO BE+ED STEP ST
520 BT1=COS(BT): BT2=SIN(BT)
530 FOR G=1 TO 2
540     FOR L=-1 TO 50 STEP ST2
550         FOR Z=-50 TO 50 STEP ST2
560             I=I+1: X=Z
570             AL=(Z+50)*2*PI/100
580             ON EL1 GOSUB 770, 790, 830, 850
590             IF G=2 THEN SWAP X,Y
600             X1=X*BT1-Y*BT2: Y1=X*BT2+Y*BT1
610             X(I)=X1*SF: Y(I)=Y1*SF
620             IF Z=-50 THEN M=M+1: PT(M)=I
630 NEXT Z,L,G,BT
640 CLS:PRINT "an="; UN; TAB(10); "P0="; P0;
650 PRINT TAB(20); "tp="; EL1; TAB(30); "ad="; AD;
660 PRINT TAB(40); "a="; A; TAB(50); "d="; D;
670 PRINT TAB(60); "st="; SF
680 FOR CX=-BX TO BX STEP UV
690     FOR CY=-BY TO BY STEP UV
700         T=1
710         FOR J=1 TO I
720             PX=CX+X(J)*SS: PY=CY+Y(J)*SS
730             ON SE1 GOSUB 870, 910, 950
740         NEXT J
750     NEXT CY,CX
760 END
770 '
780 Y=L*A*25*SIN(AL)*SIN(D*AL):RETURN
790 '
800 Y=L*A*25COS(AL)*SIN(D*AL):RETURN
810 '

```

```

820  Y=L * A * 25 * SIN(AL) * COS(D * AL):RETURN
830  '
840  Y=L * A * 25 * COS(AL) * COS(D * AL):RETURN
850  '
860  Y=0:RETURN
870  '
880  DX=320+PX: DY=200-PY
890  IF J=PT(T) THEN PSET (DX,DY): T=T+1
900  LINE-(DX,DY):RETURN
910  '
920  DX=320+PX: DY=100-PY/2
930  IF J=PT(T) THEN PSET (DX,DY): T=T+1
940  LINE-(DX,DY) :RETURN
950  '
960  DX=INT(1485+3 * PX): DY=CINT (1050+3 * PY)
970  RETURN

```

程序运行结果例如图 3.67(a), (b)所示。

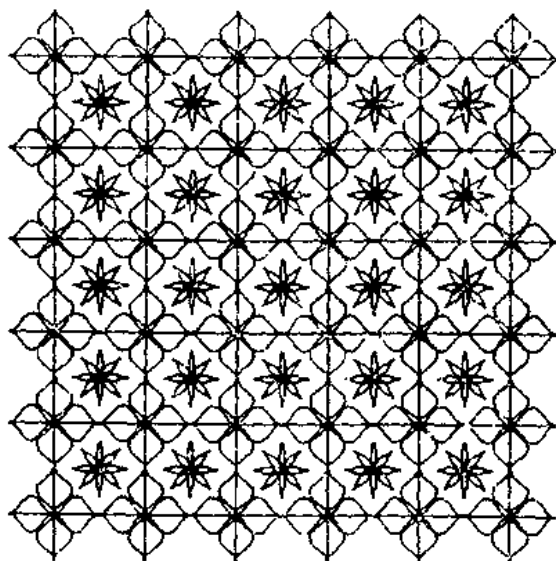


图 3.67(a)

将程序包 PC3-57 中增加一行

```
695 IF(CX>-HF1 AND CX<HF1) AND  
    (CY>-HF1 AND CY<HF1) THEN  
    GOTO 750
```

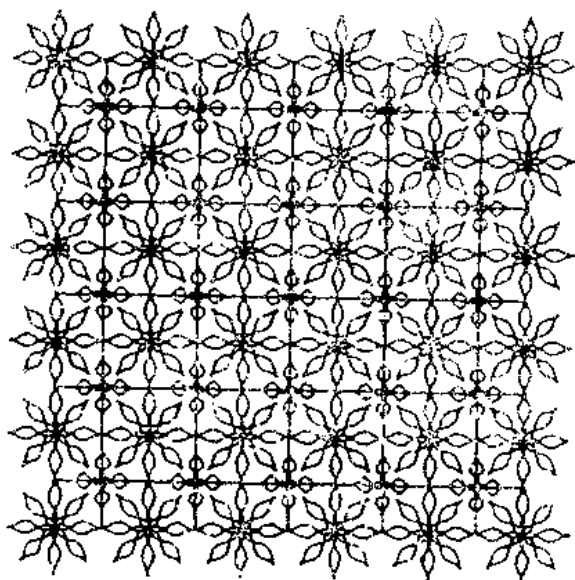


图 3.67(b)

程序中各参数取值范围不变，则可绘出各种各样的花边图案。程序运行例如图 3.68 所示。

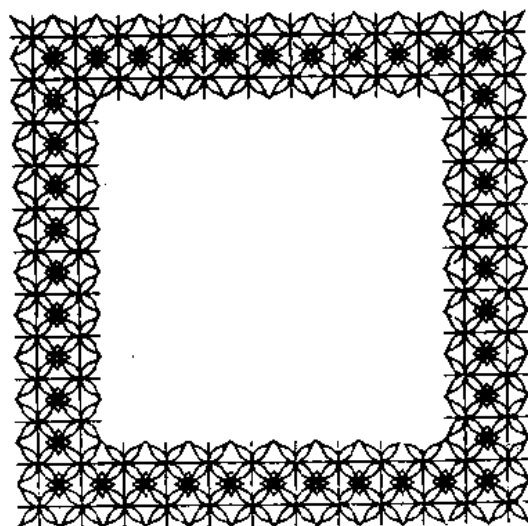


图 3.68(a)

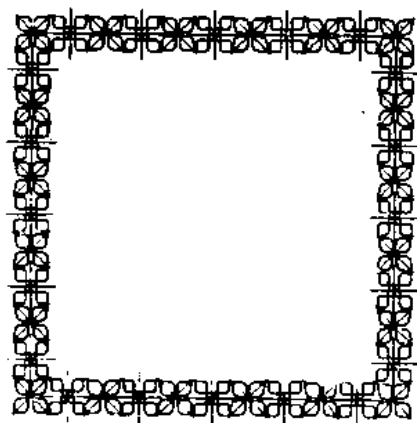


图 3.68(b)

例 3-58 花束形图案

这是一种在同心圆上配置花形单元组成的图案。

```
100 'pc3-58
110 SCREEN 2 :CLS
120 PI=3.14159
130 PX=320:PY=100:R=30:GOSUB 280
140 FOR RE=1 TO 3
150 READ STE,DI,R
160 IF RE=2 THEN LS=2*PI/STE ELSE LS=0
170 FOR A=0 TO 2*PI-LS STEP 2*PI/STE
180 PX=320+DI*COS(A):PY=200+DI*SIN(A)
185 GOSUB 280
190 NEXT A
200 NEXT RE
210 FOR A=0 TO 14*PI STEP PI/60
220 X=320+144*(1+1/5*SIN(9.060001*A))*COS(A)
230 Y=200+144*(1+1/5*SIN(9.060001*A))*SIN(A)
260 NEXT A
270 END
280 S=CINT(8-RND(1)*5)
290 IF S MOD 2=0 THEN K=2 ELSE K=1
300 FOR AN=0 TO K*PI+PI/(10*S) STEP PI/(15*S)
310 X=(R/4*SIN(3*S*AN)+R*SIN(S*AN))*COS(A
N)+PX
320 Y=(R/4*SIN(3*A*AN)+R*SIN(S*AN))*SIN(A
N)+PY
330 IF AN=0 THEN PSET(X,Y)
340 LINE-(X,Y)
350 NEXT AN:RETURN
360 DATA 7,43.6,24.5,17,77.5,18,31,102,13
```

程序中 130 行设定花束中心位置及中心花形大小数值后,转向 280~350 行的子程序画图。花型子程中,由随机函数选择花的瓣数,然后计算花的 X、Y 坐标并绘图。

140 行~200 行是双重循环。外循环变量 RE 取值 1~3 使花束图案呈现三层环状。150 行的 READ 语句读入每层环的圆半径 DI、步长分母 STE 及花型单元的直径 R 的数据。每设置一次花的位置，都转入花型子程序画图。

210~260 行循环画花束外围的环带。

程序运行结果如图 3.69。

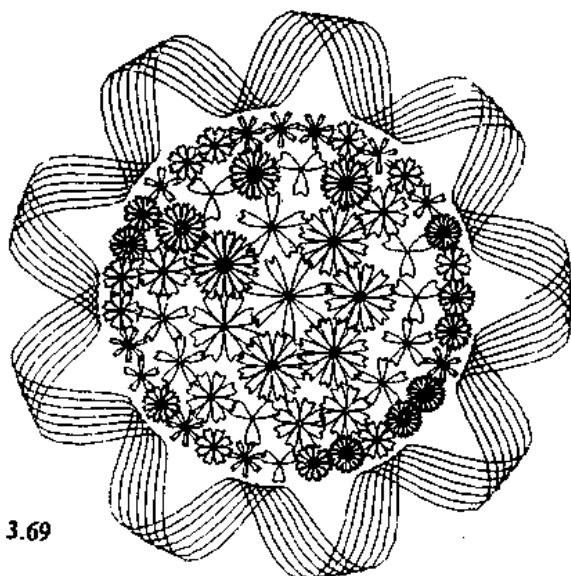


图 3.69

例 3-59 花束形图案 2

```

100 'pc3-59
110 SCREEN 2:CLS
120 PI=3.14159:D=25
130 DIM X(121),Y(121)
140 FOR TH=0 TO 2*PI+PI/60 STEP PI/60
150 I=I+1:D2=.7*D*(1+SIN(6*TH))
160 X(I)=D2*COS(TH):Y(I)=D2*SIN(TH)
170 NEXT TH
180 R=2*D:ST=PI/3:ED=2*PI:GOSUB 290

```



```

190 R = 2 * D:ST = PI / 3:ED = 2 * PI:GOSUB 290
200 R = 4 * D:COS(PI / 6):BE = PI / 6:GOSUB 290
210 R = 4 * D:BE = 0: GOSUB 290
220 R = 6 * D : GOSUB 290
230 BT = ATN(2 / 5 * SIN(PI / 3))
240 R = 5 * D / COS(BT):BE = BT:GOSUB 290
250 BE = -BT:ED = 2 * PI - BT: GOSUB 290
260 CIRCLE (320,100),200:CIRCLE (320,100),210
270 END
290 FOR A = BE TO ED STEP ST
300 PX = 320 + R * COS(A):PY = 200 + R * SIN(A)
310 FOR J = 1 TO I
320 X = PX + X(J) :Y = PY + Y(J)
330 IF J = 1 THEN PSET(X,Y / 2)
340 LINE-(X,Y / 2)
350 NEXT J,A:RETURN

```

140~170 行进行花型计算，并将计算结果存入数组变量。180~250 行配置花型，290~350 行绘图。程序运行结果如图 3.70。

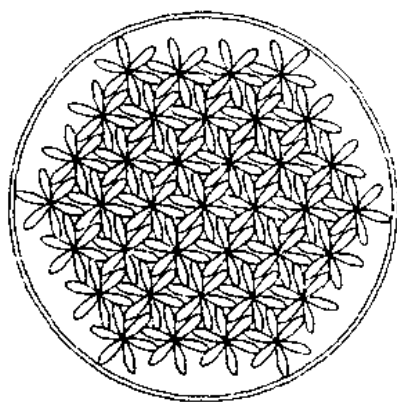


图 3.70

例 3-60 花束形图案 3

```

100 'pc3-60
110 SCREEN 2 :CLS
120 PI=3.14159:D=45
130 FOR N=1 TO 4
140 GOSUB 520
150 NEXT N
160 N=25
170 FOR D=100 TO 115 STEP 5
180 GOSUB 520
190 NEXT D
200 'ribbon.c03
210 PI=3.14159:N=25:D=180
220 FOR I=1 TO 4
230 FOR A=0 TO 2*PI STEP PI/120
240 R=D*(5/6+1/6*SIN(15*A+PI/5*I))
250 X=R/25*COS(5*A)+RCOS(A)+320
260 Y=R/25*SIN(5*A)+RSIN(A)+200
270 IF A=0 THEN PSET(X,Y/2):BX=X:BY=Y/2
280 LINE-(X,Y/2)
290 NEXT A:LINE-(BX,BY):NEXT I
300 END
310 FOR J=1 TO I
320 X=PX+X(J):Y=PY+Y(J)
330 IF J=1 THEN PSET(X,Y/2)
340 LINE-(X,Y/2)
350 NEXT J,A:RETURN
520 FOR A=0 TO 2*PI STEP PI/60
530 X=320+D/N*COS(5*A)+D*COS(A)
540 Y=200+D/N*SIN(5*A)+D*SIN(A)
550 IF A=0 THEN PSET(X,Y/2):BX=X:BY=Y/2
560 LINE-(X,Y/2)
570 NEXT A:LINE-(BX,BY)
580 RETURN

```

程序运行结果如图 3.71 所示。520~580 行为画图子程序。

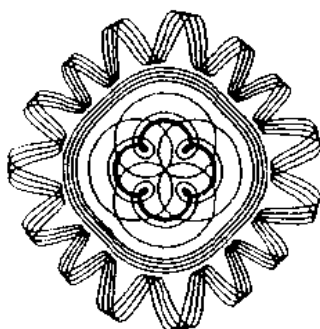


图 3.71

附：图案单元及其数学式

这里例举部分构成闭曲线图案单元的数学式及其对应的图案单元（图案单元 1~14）

数学式中角度变量 A 通常取值为 $0 \sim 2\pi$ 、 $0 \sim \pi$ 或 $0 \sim 4\pi$ ，常数 B 为图案单元的半径，常数 P 、 N 、 L 决定图案的形状。变化这些常数、可以得到不同的图案。

将数学式编到程序中，可以画出这些图案单元。

例：画出图案单元 1 的程序

```

100  'pc3-61
110  SCREEN 2:CLS
120  PI=3.14159:B=45
130  FOR PX=50 TO 450 STEP 100
140  N=N+2:L=0
150  FOR PY=40 TO 350 STEP 100
160  L=L+1
170  LOCATE 20,65:PRINT "n=";N;"l=";L

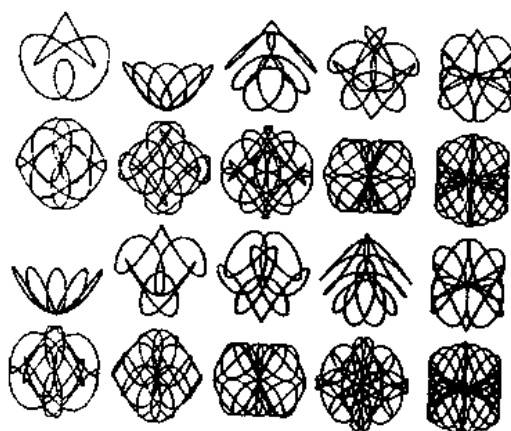
```

```

180  FOR A=0 TO 2*PI STEP PI/(30*N)
190  X=B*SIN((N)*A)*COS(5*A)+PX
200  Y=B*SIN((L+N)*A)*SIN(5*A)+PY
210  IF A=0 THEN PSET(X,Y/2):BX=X:BY=Y/2
220  LINE-(X,Y/2)
230  NEXT A:LINE-(BX,BY)
240  NEXT PY,PX
250  END

```

图案花型单元 1



$B=45$

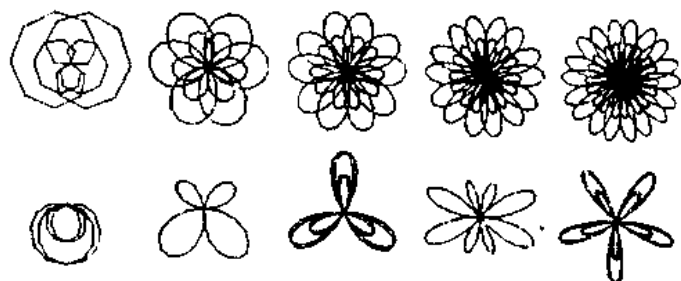
$$\begin{cases} X = B * \sin(N * A) * \cos(5 * A) \\ Y = B * \sin(L + N) * A * \sin(5 * A) \end{cases}$$

N=2, L=1	N=4, L=1	N=6, L=1	N=8, L=1	N=10, L=1
N=2, L=2	N=4, L=2	N=6, L=2	N=8, L=2	N=10, L=2
N=2, L=3	N=4, L=3	N=6, L=3	N=8, L=3	N=10, L=3
N=2, L=4	N=4, L=4	N=6, L=4	N=8, L=4	N=10, L=4

表中的 N、L 取值与图中花形单元位置相对应。

利用类似的方法，可以编出画下列图案单元的程序。

图案单元 2



$$B=35, P=0.5, A=0 \sim 2\pi$$

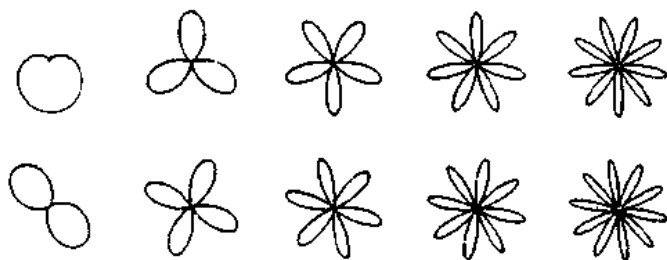
$$R = B / (3 * \sin(N * P * A) + B * \sin(N * A))$$

$$\begin{cases} X = R * \cos(2 * A) \\ Y = R * \sin(2 * A) \end{cases}$$

$$Y = R * \sin(2 * A)$$

N=1	N=3	N=5	N=7	N=9
N=2	N=4	N=6	N=8	N=10

图案单元 3



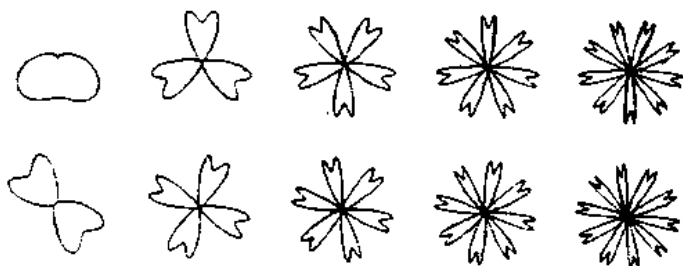
$$B = 35$$

$$R = B * (1 / 2 + 1 / 2 * \sin(N * A))$$

$$\begin{cases} X = R * \cos(A) \\ Y = R * \sin(A) \end{cases}$$

N=1	N=3	N=5	N=7	N=9
N=2	N=4	N=6	N=8	N=10

图案单元 4



$$B = 45$$

$$D = B * (4 / 5 + 1 / 5 * \sin(3 * N * A))$$

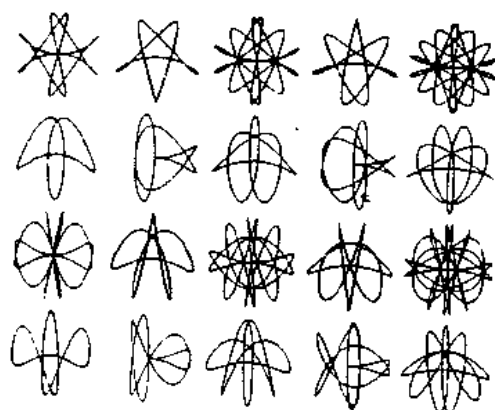
$$R = D * (1 / 2 + 1 / 2 * \sin(N * A))$$

$$\begin{cases} X = R * \cos(A) \\ Y = R * \sin(A) \end{cases}$$

N=1	N=3	N=5	N=7	N=9
N=2	N=4	N=6	N=8	N=10

程序编写参照 PC3-5。

图案单元 5

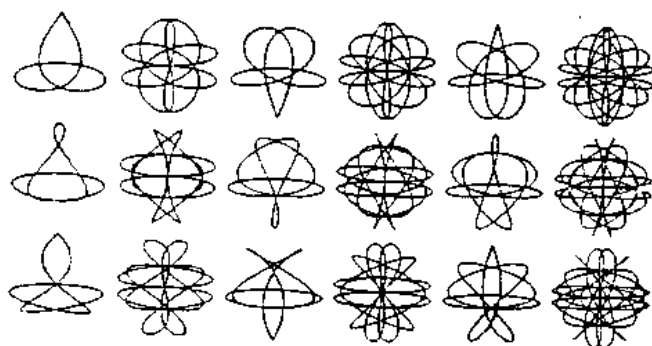


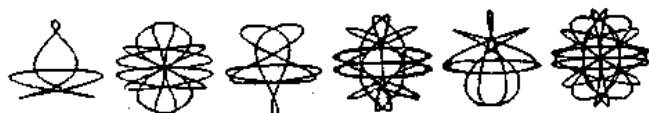
$B = 45$

$$\begin{cases} X = B * \sin(N * A) * \cos(A) \\ Y = B * \sin((N + L) * A * \sin(A)) \end{cases}$$

$N=3, L=1$	$N=4, L=1$	$N=5, L=1$	$N=6, L=1$	$N=7, L=1$
$N=3, L=2$	$N=4, L=2$	$N=5, L=2$	$N=6, L=2$	$N=7, L=2$
$N=3, L=3$	$N=4, L=3$	$N=5, L=3$	$N=6, L=3$	$N=7, L=3$
$N=3, L=4$	$N=4, L=4$	$N=5, L=4$	$N=6, L=4$	$N=7, L=4$

图案单元 6





$$B=45$$

$$\begin{cases} X = B * \sin((L + N) * A * \cos(A)) \\ Y = B * \sin(N * A) * \sin(A) \end{cases}$$

N=3, L=1	4, 1	5, 1	6, 1	7, 1	8, 1
3, 2	4, 2	5, 2	6, 2	7, 2	8, 2
3, 3	4, 3	5, 3	6, 3	7, 3	8, 3
3, 4	4, 4	5, 4	6, 4	7, 4	8, 4

第三部分 三维绘图程序设计

第四章 三维绘图入门

4.1 三维图形的平面显示

在二维平面上显示立体图形，必须进行投影变换。因而，投影基本理论是三维绘图不可缺少的基础知识。为此，首先导入表示三维图形的空间直角坐标系。

三维图形坐标系如图 4.1 所示。0 为原点，通过原点的三条互相垂直的直线，分别称为 X 轴，Y 轴和 Z 轴，并用箭头表示坐标轴的正方向。0—XY，0—YZ 和 0—XZ 分别称为 XY 平面 YZ 平面和 XZ 平面。如果在坐标轴上标上刻度，则空间任意一点 P 对应于 X、Y、Z 轴的位置，分别

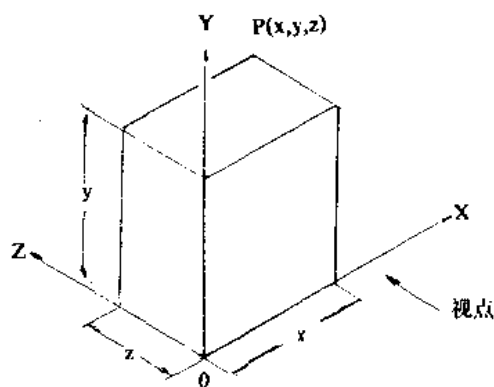


图 4.1

称为P点的X坐标和Z坐标，并记为 $P(x, y, z)$ 。

当视点和坐标系的位置关系如图4.1所示时，这种坐标系称为视点坐标系或左手坐标系。

将位于三维坐标系中的物体在二维平面上进行的显示处理，称为投影变换。例如，图4.2表示位于三维坐标系中的立体球，在视点与球之间的某个投影面上看到的图形。这时，视点可看成是人的眼睛，投影面可看成是计算机的屏幕。如果把投影面看成是透明玻璃，将眼睛在视点上看到的物体，照样画在投影面上所得到的图形，就是物体的投影图。视点和物体上各点的连线，称为投影线。

如果视点位于Z轴的反向延长线上，则视点坐标系在投影面上的投影，如图4.3所示，Z轴不显示，X轴，Y轴仍构成二维直角坐标系。

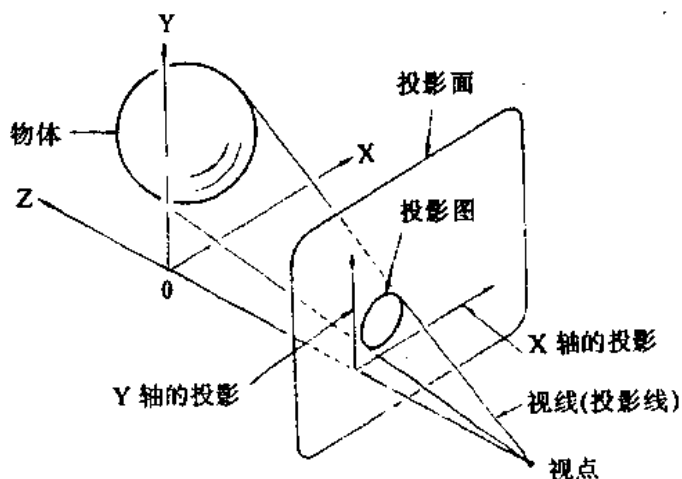


图4.2

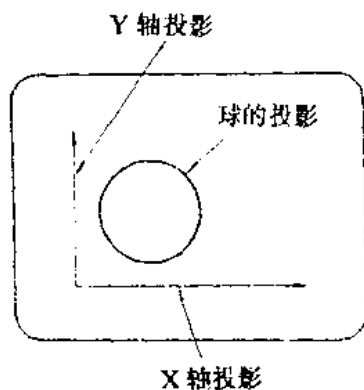


图 4.3

例如，将正方体置于左手坐标系中，当放置位置为正方体的三个基准面和坐标轴平行时，投影面上显示的图形是一个正方形。如图 4.4。

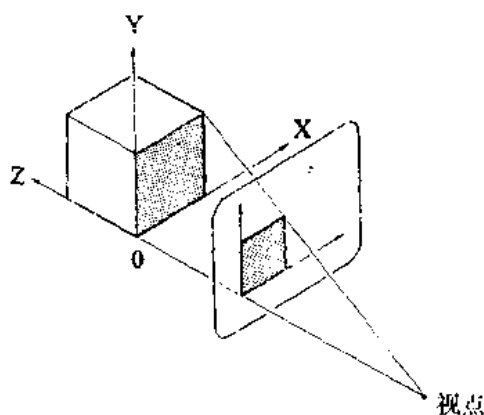


图 4.4

如果以 Y 轴为基准依右手螺旋方向将正方体旋转一个角度，这时，在投影面上可以看到立方体的两个侧面，但仍

然不能显示出立方体的样子（如图 4.5），因此，我们再将立方体绕 X 轴顺时针旋转一个角度，则如图 4.6 所示那样，投影面上显示出一个立方体。

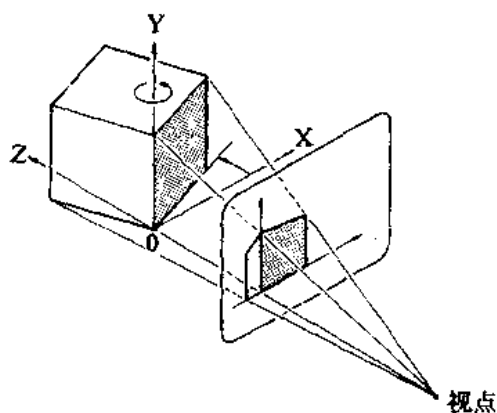


图 4.5

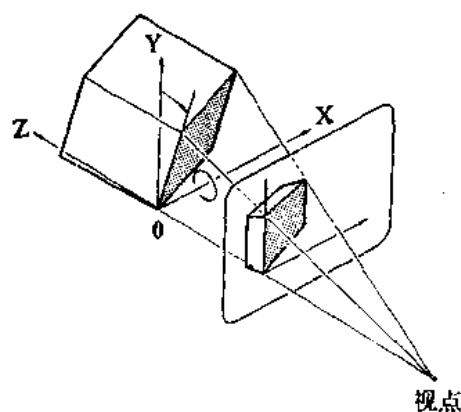


图 4.6

从图 4.6 中看出，投影面上显示出的立方体投影图中，

包围各面的棱边不是互相平行的，如图 4.7 那样，分别收集于一点。这种投影图叫做透视图，会聚点叫做灭点。根据灭点数，透视图有单灭点透视图、二灭点透视图和三灭点透视图。

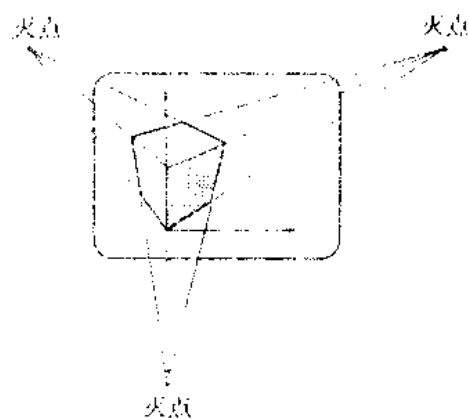


图 4.7

如果灭点位于无限远点，则视线（投影线）如图 4.8 所示，互相平行且都垂直于投影面。立方体的投影图如图 4.9 所示，各面的棱边互相平行。这种投影图称为轴测投影图。

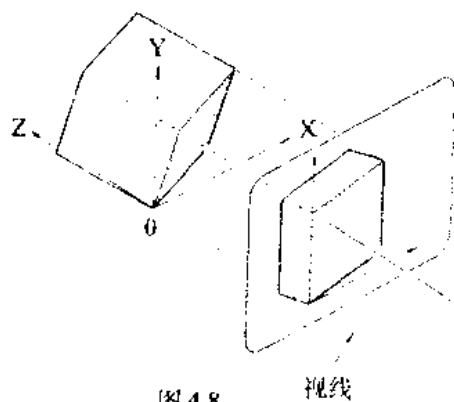


图 4.8

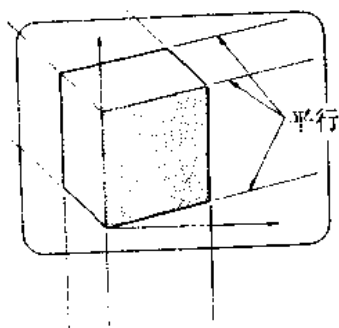


图 4.9

4.2 图形的几何变换

一、旋转变换

如图 4.10 那样，将二维平面上的点 $P(X,Y)$ 旋转 θ 角度后点 P' 的坐标为

$$X': X\cos\theta + Y\sin\theta$$

$$Y': -X\sin\theta + Y\cos\theta$$

因而，当物体一点 P 绕 Y 轴旋转 YR 角度时，点 $P_1(X_1, Y_1, Z_1)$ 的新坐标 $P_2(X_2, Y_2, Z_2)$ 中， Y 坐标值不变，即 $Y_2 = Y_1$ ，但 X 、 Z 坐标发生了变化，这时， X_2 、 Z_2 分别为：

$$X_2 = X_1 * \cos(YR) - Z_1 * \sin(YR)$$

$$Z_2 = X_1 * \sin(YR) + Z_1 * \cos(YR)$$

如图 4.11 所示

然后将点 $P_2(X_2, Y_2, Z_2)$ 绕 X 轴再旋转 XR (图 4.12)，

点 $P_3(X_3, Y_3, Z_3)$ 的坐标为:

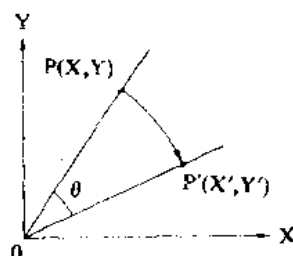


图 4.10

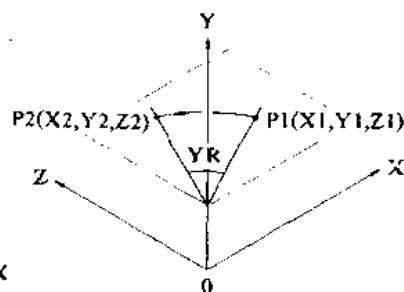


图 4.11

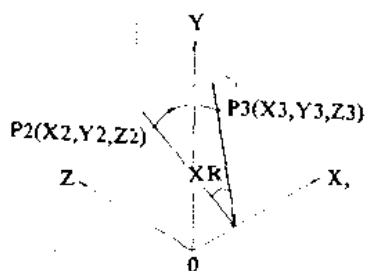


图 4.12

$$X_3 = X_2$$

$$Y_3 = Z_2 * \sin(XR) + Y_2 * \cos(XR)$$

$$Z_3 = Z_2 * \cos(XR) - Y_2 * \sin(XR)$$

如果再将点 $P_3(X_3, Y_3, Z_3)$ 绕 Z 轴旋转 ZR , 如图 4.13, 则点 $P_4(X_4, Y_4, Z_4)$ 的坐标值为。

$$Z_4 = Z_3$$

$$X_4 = Y_3 * \sin(ZR) + X_3 * \cos(ZR)$$

$$Y_4 = Y_3 * \cos(ZR) - X_3 * \sin(ZR)$$

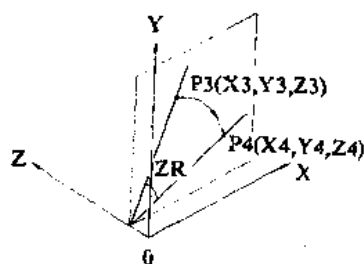


图 4.13

这样，我们看到，最初绕 Y 轴旋转是在 XZ 平面上将点 $P1(X1, Y1, Z1)$ 旋转 YR 得到点 $P2(X2, Y2, Z2)$ ；其次绕 X 轴旋转是在 YZ 平面上将点 $P2(X2, Y2, Z2)$ 旋转 XR 得到点 $P3(X3, Y3, Z3)$ ，最后绕 Z 轴旋转是在 XY 平面旋转 ZR 得到点 $P4(X4, Y4, Z4)$ 。点 $P4$ 投影在投影面上的点如图 4.14 所示，投影点的 Z 坐标消失，只显示出 $X4$ 、 $Y4$ 的值。

旋转顺序与最终图形有很大关系，即使旋转角度 $ZR = YR = XR$ ，

按绕 Y 轴 → X 轴
→ Z 轴的旋转顺序
与绕 X 轴 → Y 轴 →
Z 轴旋转后，投影
的位置和形状是不
同的。

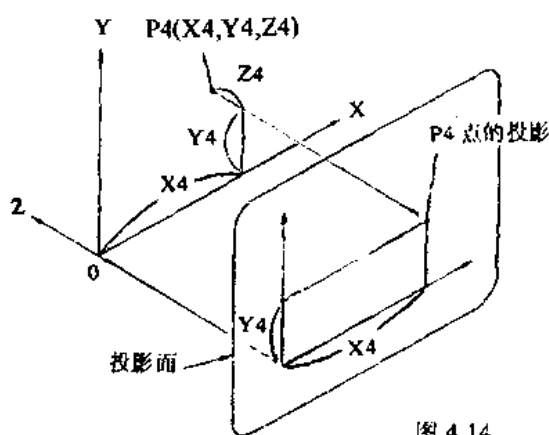


图 4.14

一般情况下，大多是依绕 Y 轴，再绕 X 轴旋转的顺序进行，从而得到立体图形，一般省略取绕 Z 轴旋转。

二、平移变换

三维图形的平移和二维图形的平移变换是相同的。例如，空间点 $P_1(X_1, Y_1, Z_1)$ 在 X 轴、Y 轴、Z 轴上的移动量分别为 X_M 、 Y_M 、 Z_M 时，则移动后的点 $P'(X_2, Y_2, Z_2)$ 的坐标为：

$$X_2 = X_1 + X_M$$

$$Y_2 = Y_1 + Y_M$$

$$Z_2 = Z_1 + Z_M$$

三、比例变换

以原点为中心，将点 $P(X_1, Y_1, Z_1)$ 的坐标值分别乘以比例系数 RT_1 、 RT_2 、 RT_3 时，则新点坐标 $P'(X_2, Y_2, Z_2)$ 分别为：

$$X_2 = X_1 \cdot RT_1$$

$$Y_2 = Y_1 \cdot RT_2$$

$$Z_2 = Z_1 \cdot RT_3$$

若 $RT_1 = RT_2 = RT_3 = RT$ 时，图形做相似变换；若 $RT_1 \neq RT_2 \neq RT_3$ 时，变换后的图形与原图形不同。

例 4-1 立方体投影图

立方体各顶点坐标如图 4.15 所示，典型程序如 PC4-1。

```
5 PC4-1
```

```
10 SCREEN 2:CLS
```

```
20 WINDOW (-320,-200)-(320,200)
```

```
30 INPUT "Yr=";YR
```

```
40 INPUT "Xr=";XR
```

```

50 INPUT "Zr=";ZR
60 P=3.14159/180
70 YR=P*YR:XR=P*XR:ZR=P*ZR
100 FOR I=1 TO 11
110 READ X1,Y1,Z1:GOSUB 200
120 IF I=1 THEN PSET (X4,-Y4) ELSE LINE -(X4,-Y4)
130 NEXT I
140 END
150 '
200 X2=X1*cos(YR)-Z1*sin(YR)
210 Y2=Y1
220 Z2=X1*sin(YR)+Z1*cos(YR)
230 X3=X2
240 Y3=Z2*sin(XR)+Y2*cos(XR)
250 Z3=Z2*cos(XR)-Y2*sin(XR)
260 X4=Y3*sin(ZR)+X3*cos(ZR)
270 Y4=Y3*cos(ZR)-X3*sin(ZR)

```

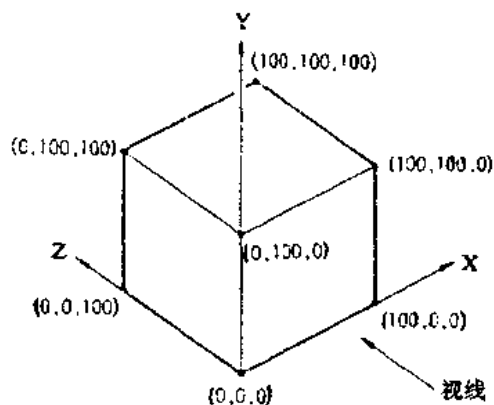


图 4.15

```
280 Z4=Z3:RETURN
```

```
300 DATA 0,0,0, 100,0,0, 100,100,0, 0,100,0, 0,0,0, 0,0,100
```

```
310 DATA 0,100,100, 0,100,0, 100,100,0, 100,100,100, 0,100,100
```

程序说明:

这是一个画立方体轴测投影图的典型程序。

10 行 设是屏幕工作在高分辨率图形方式;

20 行 设定世界坐标系窗口;

30~50 行 由键盘输入绕 X 轴、Y 轴和 Z 轴旋转的角度,单位是度。

100~130 行 输入立方体顶点数据后,转 200 行子程序进行坐标变换画图。

程序运行时,首先键入绕 X 轴、Y 轴和 Z 轴旋转的角度值,数值不同,图形各异。

4.3 透视图

一、透视原理

如图 4.16(a)所示的那样,从正上方向下看到的绕 Y 轴、X 轴旋转后的正方体的图形,是一个平面图;从横向看,是一个侧面图,如图 4.16(b)。如果将得到的平面图、侧面图各点的尺寸归纳在一个投影面上并求其对应的交点,则可得到如图 4.16(c)的正方体的透视图。

下面讨论计算方法。从图 4.17 中可以看出,立方体棱边 A^*E 在投影面上的投影 A^*E^* ,当点 E 的坐标和视点距投影面的距离 SP 为已知时,可求得 E^* 点的 Y 坐标值 Y' 为

$$Y' = \frac{Y \times SP}{Z + SP}$$

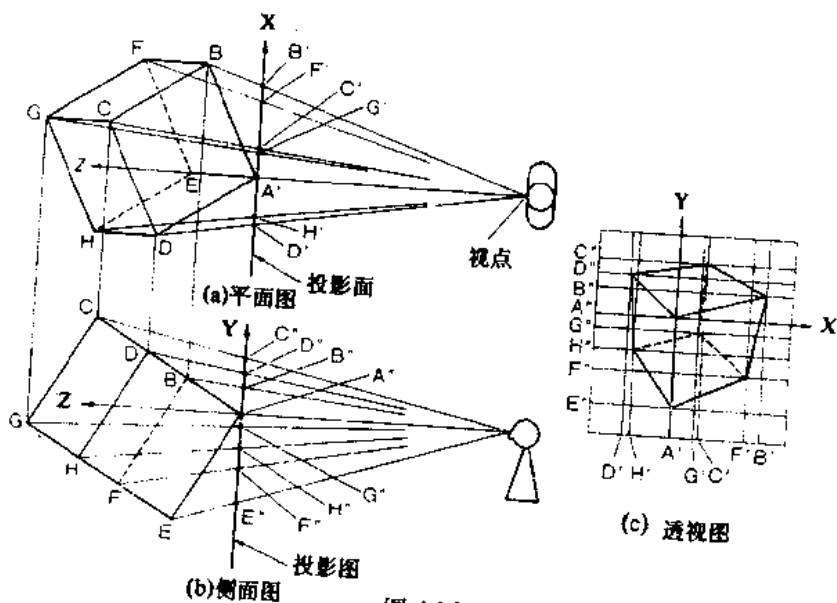


图 4.16

从而可求得 $A'E$ 的投影 $A'E'$ 的值。这个关系同样运用于物体上的其它点。这样，透视变换在程序中可表示为：

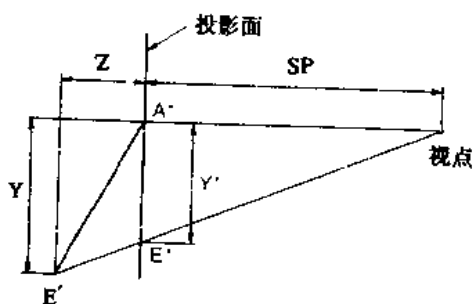


图 4.17

$$X(I) = SP * X(I) / (SP + Z(I))$$

$$Y(I) = SP * Y(I) / (SP + Z(I))$$

$$Z(I) = SP * Z(I) / (SP + Z(I))$$

二、单灭点透视图

当物体的基本面与一个坐标面平行放置时得到的透视图，称为单点透视图。例如，前述的立方体不绕 Y 轴、X 轴旋转所得到的透视图为单灭点透视图，如图 4.18。这时，灭点只有 1 个。

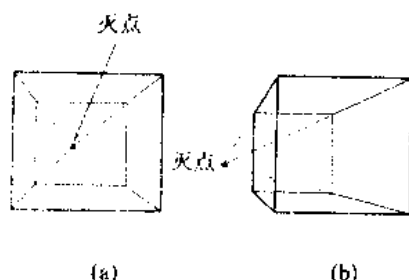


图 4.18

三、双灭点透视图

立体物体的一条棱边与三维视点坐标系三个主轴中的一条主轴平行放置时所得到的透视图，称为二灭点透视图。例如，立方体只绕 Y 轴旋转时得到透视图，是二灭点透视图，如图 4.19 所示。建筑用透视图，多采用二灭点透视图。

四、三灭点透视图

当视点坐标系的三条主轴均不平行于投影面时所得到的物体的透视图是三灭点透视图。例如，正方体绕 Y 轴、X 轴旋转后得到的透视图是三灭点透视图，如图 4.20。

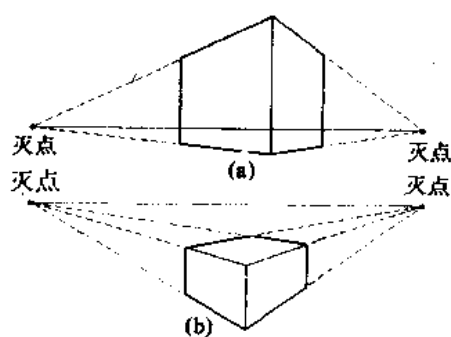


图 4.19

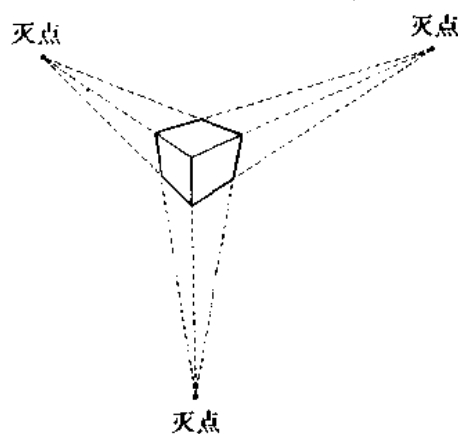


图 4.20

4.4 三维绘图基本程序包

根据前述投影变换、几何变换原理，可编成绘制三维图形的基本程序包。程序包清单如 PC4-02。

```

2000 7PC4-02
2010 '
2100 SCREEN 2:WIDTH 80:CLS
2110 POX = 640:POY = 200:PAI = 3.14159
2120 DPX = 2.678:DPY = 2.721
2150 PRINT:PRINT SPC(9)"a"
2160 PRINT SPC(9)"b"
2170 PRINT SPC(9)"c"
2180 PRINT SPC(9)"d"
2190 PRINT
2200 INPUT "a-d"; A$
2220 CLS
2230 PRINT
2240 INPUT "x0 = ,y0 = ";X0,Y0
2320 X0 = X0 * DPX:Y0 = Y0 * DPY
2330 WINDOW (-POX / 2-X0, -POY / 2+Y0)
      -(POX / 2-X0, POY / 2+Y0)
2640 PRINT:PRINT
2650 INPUT "rt = ";RT
2660 INPUT "xm = ";XM:XXM = XM
2670 INPUT "ym = ";YM:YYM = YM
2680 INPUT "zm = ";ZM:ZZM = ZM
2690 IF A$ = "d" GOTO 2760
2700 INPUT "pp-sp";SP
2710 IF A$ = "a" THEN 2770
2720 IF A$ = "b" GOTO 2740
2730 IF A$ = "c" GOTO 2750

```

```

2740 INPUT "yr=";YR:GOTO 2770
2750 INPUT "yr=",xr=";YR,XR:GOTO 2770
2760 INPUT "yr=",xr=",zr=";YR,XR,ZR
2770 CLS
2780 P=PAI/180:YR=P*YR:XR=P*XR:ZR=P*ZR
2790 RETURN
2800 '
2900 '
2910 READ NN
2920 DIM X(NN), Y(NN), Z(NN), DL$(NN)
2930 RETURN
2940 '
3000 '
3010 FOR I=1 TO NN
3020 READ X(I),Y(I),Z(I),DL$(I)
3030 NEXT I
3040 RETURN
3050 '
3100 '
3110 FOR I=1 TO NN
3120 GOSUB 3160
3130 NEXT I
3140 RETURN
3150 '
3160 X(I)=X(I)+XM:Y(I)=Y(I)+YM:Z(I)=Z(I)+ZM
3170 RETURN
3180 '

```



```

3200
3210 FOR I=1 TO NN
3220 GOSUB 3250
3230 NEXT I
3240 RETURN
3250 IF A$ = "a" THEN 3450
3260 '
3270 '
3280 X1=X(I):Z1=Z(I)
3290 X(I)=X1 * COS(YR)-Z1 * SIN(YR)
3300 Z(I)=X1 * SIN(YR)+Z1 * COS(YR)
3310 IF A$ = "b" THEN 3450
3320 '
3330 '
3340 Z2=Z(I):Y2=Y(I)
3350 Y(I)=Z2 * SIN(XR)+Y2 * COS(XR)
3360 Z(I)=Z2 * COS(XR)-Y2 * SIN(XR)
3370 IF A$ = "c" THEN 3450
3380 '
3390 '
3400 Y3=Y(I):X3=X(I)
3410 X(I)=Y3 * SIN(ZR)+X3 * COS(ZR)
3420 Y(I)=Y3 * COS(ZR)-X3 * SIN(ZR)
3430 IF A$ = "d" THEN RETURN
3440 '
3450 '
3460 X(I)=SP * X(I) / (SP+Z(I))

```

```

3470 Y(I)=SP * Y(I) / (SP+Z(I))
3480 RETURN
3490 '
3600 '
3610 FOR I=1 TO NN
3620 GOSUB 3650
3630 NEXT I
3640 RETURN
3650 '
3670 '
3680 X(I)=X(I) * DPX:Y(I)=Y(I) * DPY
3690 '
3700 X(I)=X(I) * RT:Y(I)=Y(I) * RT
3730 RETURN
3800 '
3810 FOR I=1 TO NN
3850 IF DL$(I)="p" THEN GOSUB 3900:GOTO 3870
3860 GOSUB 3940
3870 NEXT I
3880 RETURN
3890 '
3900 '
3920 PSET (X(I),-Y(I)):RETURN
3930 '
3940 '
3960 LINE-(X(I),-Y(I)):RETURN
3980 '

```

程序包由下列子程序构成:

(1) 初始化、投影方式选择子程序 (2100~2790 行),

2100 行 设定屏幕工作在高分辨率图形方式(640×200)

2150~2220 行 投影图方式选择:

A 单灭点透视投影

B 2 灭点透视投影

C 3 灭点透视投影

D 轴测投影

由键盘输入 A~D 进行方式选择.

2240 行 选择坐标原点是否移动, 否则取坐标系原点
为屏幕中心. 由键盘输入平移量值 X0, Y0.

2330 行 设置窗口; 2650 行 输入比例参数 RT.

2660 行 输入平行移动变换参数:

XM=?

YM=?

ZM=?

由键盘输入各参数数据;

2700 行 由键盘输入透视投影中视点距离 SP;

2740~2760 行 键入图形旋转变换参数:

YR;

YR, XR

YR, XR, ZR

(2) 数组说明子程序(2910~2930 行).

2910 行 指定数据数;

2920 行 定义数组

(3) 读入数据子程序(3000~3040 行).

(4) 平移变换子程序(3100~3170 行).

(5)旋转变换、透视变换子程序(3200~3480 行)。

3270~3300 行 绕 Y 轴旋转;

3330~3360 行 绕 X 轴旋转;

3400~3430 行 绕 Z 轴旋转;

3450~3480 行 透视变换子程序;

(6)比例变换子程序序(3600~3730 行)。

(7)画图子程序(3800~3900 行)。

下章为三维绘图程序是此基本程序包的应用实例。

第五章 三维绘图程度 17 例

例 5-1 圆柱图案

圆柱是最基本的几何图案之一。在屏幕上可以画出沿不同坐标轴方向的圆柱图案。

1、沿 X 轴方向的圆柱。

程序如 PC5-1

```
5 'PC5-1(1)
10 GOSUB 2100: GOSUB 2900: GOSUB 4000: GOSUB 4100:
   GOSUB 4200
20 RESTORE 120
30 GOSUB 4000: XXC = XC: XC = XC + 50: GOSUB 4100:
   GOSUB 4200
40 FOR J = 0 TO 2 * PAI STEP PAI / 12
50 XC = XXC: X1 = XC: Y1 = R * SIN(J) + YC:
   Z1 = R * COS(J) + ZC
60 X(I) = X1: Y(I) = Y1: Z(I) = Z1: ZZ1 = Z1
70 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB 3900
80 X1 = XC + 50: X(I) = X1: Y(I) = Y1: Z(I) = ZZ1
90 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB 3940
100 NEXT J
110 DATA 100
120 DATA 20,30,0,0,0,360
950 END
4000 'read data
```

```

4010 READ R,XC,YC,ZC,AS,AE
4020 RETURN
4030 /
4100 /
4110 N=INT(R * RT / 1.5+20): BB=2 * PAI / N
4120 AS=P * AS: AE=P * AE
4130 RETURN
4140 /
4200 /
4210 FOR I=INT(AS / BB) TO INT(AE / BB)+1
4220 AA=BB * I
4230 IF I=INT(AS / BB) THEN AA=AS
4240 IF I=INT(AE / BB)+1 THEN AA=AE
4250 YC2=R * SIN(AA): ZC2=R * COS(AA)
4260 X(I)=XC: Y(I)=YC+YC2: Z(I)=ZC+ZC2
4270 GOSUB 3160: GOSUB 3250: GOSUB 3650
4280 IF AA=AS THEN GOSUB 3900: GOTO 4300
4290 GOSUB 3940
4300 NEXT I
4310 RETURN

```

说明

10~950 行 画圆柱主程序。主程序中除调用三维图形基本程序包 (PC4-2) 中的有关子程序外, 还追加了在 X 面上画圆的子程序。

4000~4020 行 读入画圆参数值的子程序。R 为圆半径, XC, YC, ZC 是圆心的空间坐标值, AS 是画圆起始角, AE 为画圆终止角。

4100~4130 行 将画圆参数一般化的子程序。

4200~4310 行 在 X 面上画圆的子程序。

所谓 X 面上的圆，是指在视点坐标系中，将在 YZ 平面上所画的圆依绕 Y 轴、X 轴顺序进行旋转后所得图形的投影图；如图 5.1 所示。

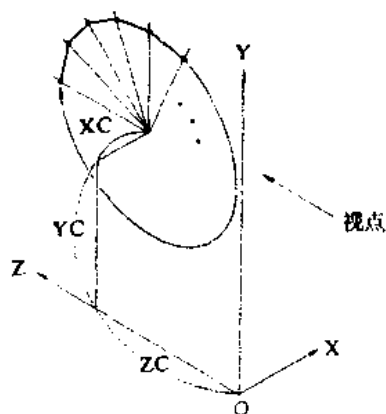


图 5.1

程序执行时，由 10 行读入 120 行所给定的画圆的参考数值；20 行重读各画圆的参数值，并由 30 行语句中将 XC 坐标值加上 50，以确定新的圆心坐标值。分别画出圆柱的两个底圆。最后由 40~100 行画圆柱的侧面线。

当键盘输入各参数为：

D	轴测投影图
X0,Y0	0,0
RT	1
XM	0
YM	0
ZM	0
XR,YR,ZR	25,-35.266,0

时，屏幕上显示如图 5.2.

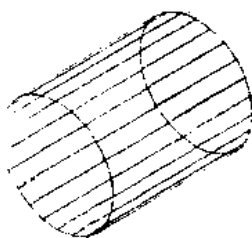


图 5.2

2、星状分布的圆柱图案

程序如 PC5-1(2)

```
5 'pc5-1(2)
10 GOSUB 2100: GOSUB 2900: GOSUB 4000: GOSUB 4100:
   GOSUB 4200
20 RESTORE 120
30 GOSUB 4000: XXC=XC: XC=XC+50: GOSUB 4100:
   GOSUB 4200
40 FOR J=0 TO 2*PAI STEP PAI/12
50 XC=XXC: X1=XC: Y1=R*SIN(J)+YC:
   Z1=R*COS(J)+ZC
60 X(I)=X1: Y(I)=Y1: Z(I)=Z1: ZZ1=Z1
70 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB 3900
80 X1=XC+50: X(I)=X1: Y(I)=Y1: Z(I)=ZZ1
90 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB 3940
110 DTAT 100
```



```

120 DATA 20,30,0,0,0,360
200 GOSUB 4000:GOSUB 4100:GOSUB 4400
210 RESTORE 300
220 GOSUB 4000:YYC=YC:YC=YC-50:GOSUB 4100:
    GOSUB 4400
230 FOR J=0 TO 2*PAI STEP PAI/12
240 YC=YYC:X1=R*COS(J)+XC:Y1=YC:Z1=R*SIN(J)+ZC
250 X(I)=X1:Y(I)=Y1:Z(I)=Z1:XX1=X1:YY1
    =Y1:ZZ1=Z1
260 GOSUB 3160:GOSUB 3260:GOSUB 3650:GOSUB 3900
270 YY1=YC-50:X(I)=XX1:Y(I)=YY1:Z(I)=ZZ1
280 GOSUB 3160:GOSUB 3260:GOSUB 3650:GOSUB 3940
290 NEXT J
300 DATA 20,0,-30,0,0,360
310 /
400 GOSUB 4000:GOSUB 4100:GOSUB 4600
410 RESTORE 500
420 GOSUB 4000:ZZC=ZC:ZC=ZC+50:GOSUB
    4100:GOSUB 4600
430 FOR J=0 TO 2*PAI STEP PAI/12
440 ZC=ZZC:X1=R*COS(J)+XC:Y1=RSIN(J)+YC:Z1=ZC
450 X(I)=X1:Y(I)=Y1:Z(I)=Z1:XX1=X1:YY1=Y1:ZZ1=Z1
460 GOSUB 3160:GOSUB 3260:GOSUB 3650:GOSUB 3900
470 ZZ1=ZC+50:X(I)=XX1:Y(I)=YY1:Z(I)=ZZ1
480 GOSUB 3160:GOSUB 3260:GOSUB 3650:GOSUB 3940
490 NEXT J
500 DATA 20,0,0,30,0,360

```

```

4000 'read data
4010 READ R,XC,YC,ZC,AS,AE
4020 RETURN
4030 '
4100 '
4110 N=INT(R*RT/1.5+20):BB=2*PI/N
4120 AS=P*AS:AE=P*AE
4130 RETURN
4140 '
4200 '
4210 FOR I=INT(AS/BB) TO INT(AE/BB)+1
4220 AA=BB*I
4230 IF I=INT(AS/BB) THEN AA=AS
4240 IF I=INT(AE/BB)+1 THEN AA=AE
4250 YC2=R*SIN(AA):ZC2=R*COS(AA)
4260 X(I)=XC:Y(I)=YC+YC2:Z(I)=ZC+ZC2
4270 GOSUB 3160:GOSUB 3250:GOSUB 3650
4280 IF AA=AS THEN GOSUB 3900:GOTO 4300
4290 GOSUB 3940
4300 NEXT I
4310 RETURN
4320 '
4400 '
4410 FOR I=INT(AS/BB) TO INT(AE/BB)+1
4420 AA=BB*I
4430 IF I=INT(AS/BB) THEN AA=AS
4440 IF I=INT(AE/BB)+1 THEN AA=AE

```

```

4450 ZC2 = R * SIN(AA): XC2 = R * COS(AA)
4460 X(I) = XC + XC2: Y(I) = YC: Z(I) = ZC + ZC2
4470 GOSUB 3160: GOSUB 3260: GOSUB 3650
4480 IF AA = AS THEN GOSUB 3900: GOTO 4500
4490 GOSUB 3940
4500 NEXT I
4510 RETURN
4600 /
4610 FOR I = INT(AS / BB) TO INT(AE / BB) + 1
4620 AA = BB * I
4630 IF I = INT(AS / BB) THEN AA = AS
4640 IF I = INT(AE / BB) + 1 THEN AA = AE
4650 YC2 = R * sin(AA): XC2 = R * cos(AA)
4660 X(I) = XC + XC2: Y(I) = YC + YC2: Z(I) = ZC
4670 GOSUB 3160: GOSUB 3260: GOSUB 3650
4680 IF AA = AS THEN GOSUB 3900: GOTO 4700
4690 GOSUB 3940
4700 NEXT I
4710 RETURN

```

说明:

在屏幕上显示沿空间三维坐标三个坐标轴方向分布有三个圆柱, 则形成星状分布的圆柱图案。

10~950 行 绘图主程序。在 PC5-1 基础上追加了 4400~4510 画 Y 面圆和 4600~4710 画 Z 面圆两个子程序。

当程序运行中, 键入参数为:

C	三点透视图
X0,Y0	0,0
RT	1
XM	0
YM	0
ZM	0
SP	250
YR,XR	25,30

时，屏幕显示图形如图 5.3 所示。

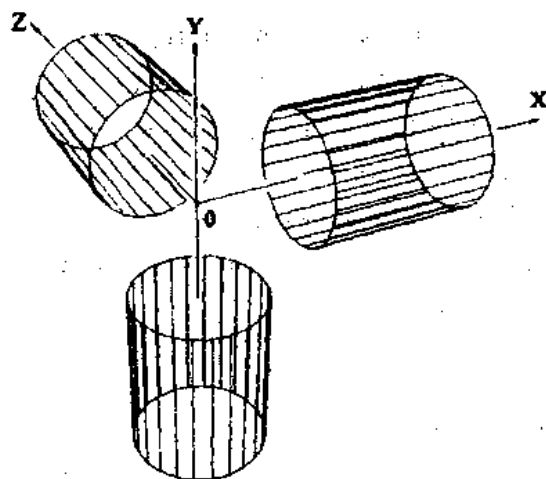


图 5.3

变化键入参数，可显示不同的图案。

例 5-2 圆锥图案

将画圆柱图案程序中圆柱一端的 R 设为 0，作为圆锥顶点，即可画出圆锥图案。画圆锥图案的程序如下：

```

5 'pc5-2
10 GOSUB 2100: GOSUB 2900: GOSUB 4000: GOSUB 4100:
   GOSUB 4200
20 RESTORE 120
30 GOSUB 4000: XXC = XC: RR = R: R = 0: XC = XC + 50:
   GOSUB 4100: GOSUB 4200
40 FOR J = 0 TO 2 * PAI STEP PAI / 12
50 R = RR: XC = XXC: X1 = XC: Y1 = R * SIN(J) + YC:
   Z1 = R * COS(J) + ZC
60 X(I) = X1: Y(I) = Y1: Z(I) = Z1: ZZ1 = Z1
70 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB 3900
80 X1 = XC + 50: X(I) = X1: Y(I) = YC: Z(I) = ZC
90 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB 3940
100 NEXT J
110 DATA 100
120 DATA 20,30,0,0,0,360
950 END
4000 'read data
4010 READ R, XC, YC, ZC, AS, AE
4020 RETURN
4030 '
4100 '
4110 N = INT(R * RT / 1.5 + 20): BB = 2 * PAI / N
4120 AS = P * AS: AE = P * AE
4130 RETURN
4140 '
4200 '

```

```

4210 FOR I=INT(AS / BB) TO INT(AE / BB)+1
4220 AA=BB * I
4230 IF I=INT(AS / BB) THEN AA=AS
4240 IF I=INT(AE / BB)+1 THEN AA=AE
4250 YC2=R * SIN(AA): ZC2=R * COS(AA)
4260 X(I)=XC: Y(I)=YC+YC2: Z(I)=ZC+ZC2
4270 GOSUB 3160: GOSUB 3250: GOSUB 3650
4280 IF AA=AS THEN GOSUB 3900: GOTO 4300
4290 GOSUB 3940
4300 NEXT I
4310 RETURN

```

程序运行时，如键入绘图参数为

D	轴测投影图
X0,Y0	0,0
RT	1
XM	0
YM	0
ZM	0
YR,XR,ZR	45,35.266,0

运行结果如图 5.4 所示。

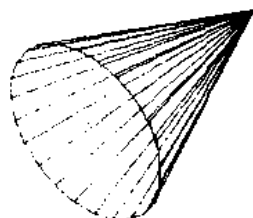


图 5.4

参照 PC5-1 编写程序, 可绘出星状分布的圆锥图案; 进行以 X 轴、Y 轴和以原点为对称的对称变换, 可绘出丰富的图案.

例 5-3 圆锥台图案

绘圆锥台图案的程序如下:

```
5 'pc5-3
10 GOSUB 2100: GOSUB 2900: GOSUB 4000: GOSUB 4100:
   GOSUB 4200
20 RESTORE 120
30 GOSUB 4000: X1=XC: R1=R: R=10: XC=XC+50:
   GOSUB 4100: GOSUB 4200
40 FOR J=0 TO 2*PAI STEP PAI/12
50 R=R1: XC=X1: X1=XC: Y1=R*SIN(J)+YC:
   Z1=R*COS(J)+ZC
60 X(I)=X1: Y(I)=Y1: Z(I)=Z1: ZZ1=Z1
70 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB 3900
80 X1=XC+50: X(I)=X1: R=10:
   Y(I)=R*SIN(J)+YC: Z(I)=R*COS(J)+ZC
90 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB 3940
100 NEXT J
110 DATA 100
120 DATA 20,30,0,0,0,360
130 END
4000 'read data
4010 READ R,XC,YC,ZC,AS,AE
4020 RETURN
4030 '
```

```

4100 /
4110 N=INT(R * RT / 1.5+20): BB=2 * PAI / N
4120 AS=P * AS: AE=P * AE
4130 RETURN
4140 /
4200 /
4210 FOR I=INT(AS / BB) TO INT(AE / BB)+1
4220 AA=BB * I
4230 IF I=INT(AS / BB) THEN AA=AS
4240 IF I=INT(AE / BB)+1 THEN AA=AE
4250 YC2=R * SIN(AA): ZC2=R * COS(AA)
4260 X(I)=XC: Y(I)=YC+YC2: Z(I)=ZC+ZC2
4270 GOSUB 3160: GOSUB 3250: GOSUB 3650
4280 IF AA=AS THEN GOSUB 3900: GOTO 4300
4290 GOSUB 3940
4300 NEXT I
4310 RETURN

```

程序运行后键入参数为

D	轴测投影
X0,Y0	0,0
RT	1
XM	0
YM	0
ZM	0
YR,XR,ZR	20,35.266,0

屏幕显示图形为图 5.5 所示。

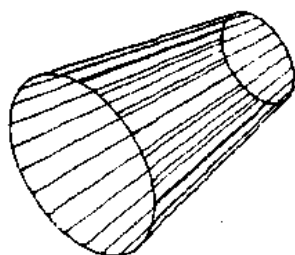


图 5.5

同样，参照 PC5-3 修改本程序，可绘出星状分布的圆锥台图案。

例 5-4 立体球图案

立体球有各种不同的表示方式。例如，用径线可表示一个球，用纬线也可以画一个球。同时，球可以绕各坐标轴旋转，从而绘出不同的图案。

1、用径线表示的星状分布立体球图案。

程序如 PC5-4 (1)

```

5 'pc5-4(1)
10 GOSUB 2100: GOSUB 2900
20 READ DVN,R,XC,YC,ZC,AS,AE
30 GOSUB 4100: GOSUB 4800
40 DATA 100
50 DATA 10,40,65,0,0,0,360
100 READ DVN,R,XC,YC,ZC,AS,AE
110 GOSUB 4100: GOSUB 5000
150 DATA 10,40,0,-65,0,0,360
    
```

160 '

200 READ DVN,R,XC,YC,ZC,AS,AE

210 GOSUB 4100: GOSUB 5200

250 DATA 10,40,0,0,65,0,360

950 END

4000 'read data

4010 READ R,XC,YC,ZC,AS,AE

4020 RETURN

4030 '

4100 '

4110 N=INT(R * RT / 1.5+20): BB=2 * PAI / N

4120 AS=P * AS: AE=P * AE

4130 RETURN

4140 '

4200 '

4210 FOR I=INT(AS / BB) TO INT(AE / BB)+1

4220 AA=BB * I

4230 IF I=INT(AS / BB) THEN AA=AS

4240 IF I=INT(AE / BB)+1 THEN AA=AE

4250 YC2=R * SIN(AA): ZC2=R * COS(AA)

4260 X(I)=XC: Y(I)=YC+YC2: Z(I)=ZC+ZC2

4270 GOSUB 3160: GOSUB 3250: GOSUB 3650

4280 IF AA=AS THEN GOSUB 3900: GOTO 4300

4290 GOSUB 3940

4300 NEXT I

4310 RETURN

4320 '

```

4800 '
4810 FOR SXR=0 TO PAI STEP PAI / DVN
4820 FOR I=INT(AS / BB) TO INT(AE / BB)+1
4830 IF I=INT(AS / BB) THEN AA=AS: GOTO 4860
4840 IF I=INT(AE / BB)+1 THEN AA=AE: GOTO 4860
4850 AA=BB * I
4860 ZC2=R * SIN(AA): XC2=R * COS(AA)
4870 X(I)=XC2: Y(I)=0: Z(I)=ZC2
4880 Y(I)=Z(I) * SIN(S * R)+YC: Z(I)=Z(I) * COS(SXR)+ZC
      X(I)=X(I)+XC
4890 GOSUB 3160: GOSUB 3260: GOSUB 3650
4900 IF AA=AS THEN GOSUB 3900: GOTO 4920
4910 GOSUB 3940
4920 NEXT I
4930 NEXT SXR
4940 RETURN
4950 '
5000 '
5010 FOR SZR=0 TO PAI STEP PAI / DVN
5020 FOR I=INT(AS / BB) TO INT(AE / BB)+1
5030 IF I=INT(AS / BB) THEN AA=AS: GOTO 5060
5040 IF I=INT(AE / BB)+1 THEN AA=AE: GOTO 5060
5050 AA=BB * I
5060 YC2=R * SIN(AA): XC2=R * COS(AA)
5070 X(I)=XC2: Y(I)=YC2: Z(I)=0
5080 Z(I)=X(I) * COS(SZR)+ZC: X(I)=X(I) * SIN(SZR)+XC:
      Y(I)=Y(I)+YC

```

```

5090 GOSUB 3160: GOSUB 3260: GOSUB 3650
5100 IF AA=AS THEN GOSUB 3900: GOTO 5120
5110 GOSUB 3940
5120 NEXT I
5130 NEXT SZR
5140 RETURN
5150 /
5200 /
5210 FOR SYR=0 TO PAI STEP PAI/DVN
5220 FOR I=INT(AS/BB) TO INT(AE/BB)+1
5230 IF I=INT(AS/BB) THEN AA=AS: GOTO 5260
5240 IF I=INT(AE/BB)+1 THEN AA=AE: GOTO 5260
5250 AA=BB*I
5260 YC2=R*SIN(AA): ZC2=R*COS(AA)
5270 X(I)=0: Y(I)=YC2: Z(I)=ZC2
5280 X(I)=Y(I)*SIN(SYR)+XC: Y(I)=Y(I)*COS(SYR)+YC:
      Z(I)=Z(I)+ZC
5290 GOSUB 3160: GOSUB 3260: GOSUB 3650
5300 IF AA=AS THEN GOSUB 3900: GOTO 5320
5310 GOSUB 3940
5320 NEXT I
5330 NEXT SYR
5340 RETURN

```

说明:

10~950 行 画图主程序。除继续调用前述 4000~4320 画图子程序外，程序中追加了 4800~4940，5000~5150，5200~5340 三个画球子程序。

程序运行时，键入参数：

D	轴测投影图
X0,Y0	0,0
RT	0.7
XM	0
YM	0
ZM	0
YR,XR,ZR	45,30,20

运行结果如图 5.6 所示。

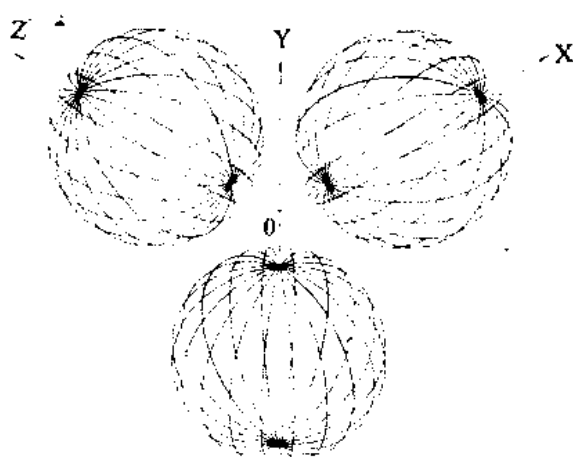


图 5.6

2、用径线表示的半球。

程序如 PC5-4 (2)

5 'pc5-4(2)

```

10 GOSUB 2100: GOSUB 2900
20 READ DVN,R,XC,YC,ZC,AS,AE
30 GOSUB 4100: GOSUB 4800
50 DATA 100
60 DATA 80,50,0,0,0,220,500
950 END

4000 'read data
4010 READ R,XC,YC,ZC,AS,AE
4020 RETURN
4030 '
4100 '
4110 N=INT(R * RT / 1.5+20): BB=2 * PAI / N
4120 AS=P * AS: AE=P * AE
4130 RETURN
4800 '
4810 FOR SXR=0 TO PAI STEP PAI / DVN
4820 FOR I=INT(AS / BB) TO INT(AE / BB)+1
4830 IF I=INT(AS / BB) THEN AA=AS: GOTO 4860
4840 IF I=INT(AE / BB)+1 THEN AA=AE: GOTO 4860
4850 AA=BB * I
4860 ZC2=R * SIN(AA): XC2=R * COS(AA)
4870 X(I)=XC2: Y(I)=0: Z(I)=ZC2
4880 Y(I)=Z(I) * SIN(SXR)+YC: Z(I)=Z(I) * COS(SXR)+ZC:
      X(I)=X(I)+XC
4890 GOSUB 3160: GOSUB 3260: GOSUB 3650
4900 IF AA=AS THEN GOSUB 3900: GOTO 4920
4910 GOSUB 3940

```

4920 NEXT I

4930 NEXT SXR

4940 RETURN

程序运行时键入参数为

C	三点透视图
X0,Y0	0,0
RT	0.5
XM	0
YM	0
ZM	0
SP	100
YR,XR	45,35

运行结果如图 5.7 所示。

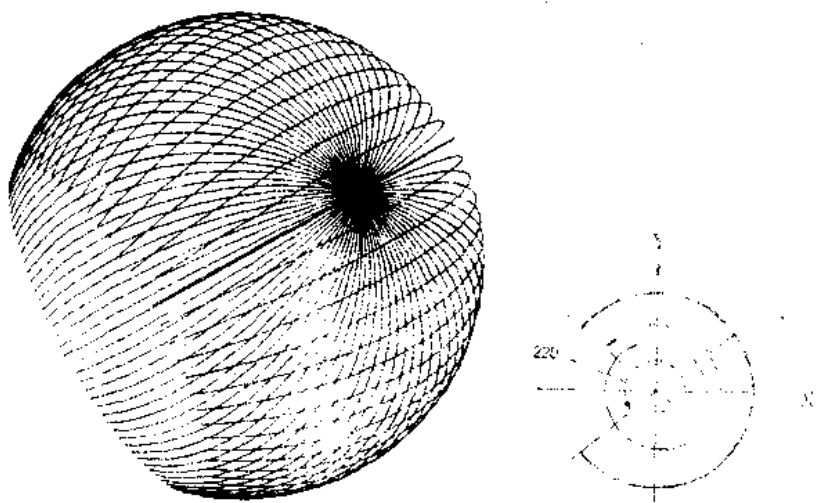


图 5.7

3、纬线星状分布球图案：

程序如 PC5-4 (3)。

```
5 'pc5-4(3)
10 GOSUB 2100: GOSUB 2900
20 READ DVN,R,XC,YC,ZC,AS,AE
30 GOSUB 5350
150 DATA 100
160 DATA 10,40,65,0,0,0,360
200 RESTORE 300: READ DVN,R,XC,YC,ZC,AS,AE
210 GOSUB 5370
300 DATA 10,40,0,-65,0,0,360
310 '
400 RESTORE 500: READ DVN,R,XC,YC,ZC,AS,AE
410 GOSUB 5390
500 DATA 10,40,0,0,65,0,360
950 END
4000 'read data
4010 READ R,XC,YC,ZC,AS,AE
4020 RETURN
4200 '
4210 FOR I=INT(AS/BB) TO INT(AE/BB)+1
4220 AA=BB*I
4230 IF I=INT(AS/BB) THEN AA=AS
4240 IF I=INT(AE/BB)+1 THEN AA=AE
4250 YC2=R*SIN(AA): ZC2=R*COS(AA)
4260 X(I)=XC: Y(I)=YC+YC2: Z(I)=ZC+ZC2
```



```

4270 GOSUB 3160: GOSUB 3250: GOSUB 3650
4280 IF AA = AS THEN GOSUB 3900: GOTO 4300
4290 GOSUB 3940
4300 NEXT I
4310 RETURN
4320 '
4400 '
4410 FOR I=INT(AS / BB) TO INT(AE / BB)+1
4420 AA = BB * I
4430 IF I=INT(AS / BB) THEN AA = AS
4440 IF I=INT(AE / BB)+1 THEN AA = AE
4450 ZC2 = R * SIN(AA):XC2 = R * COS(AA)
4460 X(I) = XC+XC2: Y(I) = YC: Z(I) = ZC+ZC2
4470 GOSUB 3160: GOSUB 3260: GOSUB 3650
4480 IF AA = AS THEN GOSUB 3900: GOTO 4500
4490 GOSUB 3940
4500 NEXT I
4510 RETURN
4600 '
4610 FOR I=INT(AS / BB) TO INT(AE / BB)+1
4620 AA = BB * I
4630 IF I=INT(AS / BB) THEN AA = AS
4640 IF I=INT(AE / BB)+1 THEN AA = AE
4650 YC2 = R * SIN(AA): XC2 = R * COS(AA)
4660 X(I) = XC+XC2: Y(I) = YC+YC2: Z(I) = ZC
4670 GOSUB 3160: GOSUB 3260: GOSUB 3650
4680 IF AA = AS THEN GOSUB 3900: GOTO 4700

```

```

4690 GOSUB 3940
4700 NEXT I
4710 RETURN
5350 '
5352 RR=R:XXC=XC:AS=P*AS:AE=P*AE
5354 FOR ANGX=-PAI/2+PAI/DVN TO
    PAI/2-PAI/DVN STEP PAI/DVN
5356 R=RR*COS(ANGX):N=INT(R*RT/1.5+20)
    BB=2*PAI/N
5358 XC=XXC+RR*SIN(ANGX):GOSUB 4200
5360 NEXT ANGX:RETURN
5370 '
5372 RR=R:YYC=YC:AS=P*AS:AE=P*AE
5374 FOR ANGY=-PAI/2+PAI/DVN TO
    PAI/2-PAI/DVN STEP PAI/DVN
5376 R=RR*COS(ANGY):N=INT(R*RT/1.5+20)
    BB=2*PAI/N
5378 YC=YYC+RR*SIN(ANGY):GOSUB 4400
5380 NEXT ANGY:RETURN
5382 '
5390 '
5392 RR=R:ZZC=ZC:AS=P*AS:AE=P*AE
5394 FOR ANGZ=-PAI/2+PAI/DVN TO
    PAI/2-PAI/DVN STEP PAI/DVN
5396 R=RR*COS(ANGZ):N=INT(R*RT/1.5+20)
    BB=2*PAI/N
5398 ZC=ZZC+RR*SIN(ANGZ):GOSUB 4600

```

5399 NEXT ANGZ: RETURN

程序运行键入参数为:

D	轴测投影图
X0,Y0	0,0
RT	0.3
XM	0
YM	0
ZM	0
YR,XR,ZR	20,30,30

屏幕显示图表如图 5.8 所示。

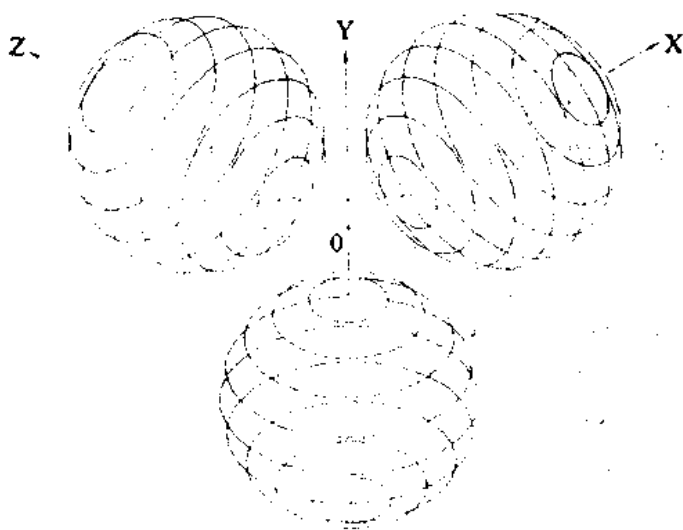


图 5.8

例 5-5 并进构成的图案

在不同位置重复绘图画相同图形而构成的图案，称为并

进图案。

1、立方体并进图案。

立方体并进图案程序如下：

```
5 'pc5-5(1)
10 GOSUB 2100:GOSUB 2900
15 XM=0:YM=0:ZM=0
20 FOR YM=0 TO 80 STEP 20
30 FOR XM=0 TO 80 STEP 20
40 RESTORE 90:GOSUB 3000:GOSUB 3100:GOSUB 5400
50 GOSUB 3200:GOSUB 3600:GOSUB 3800
60 NEXT XM
70 NEXT YM
80 DATA 11
90 DATA 0,0,0,p,10,0,0,*,10,10,0,*,10,10,10,*,
      0,10,10,*,0,0,10,*,0,0,0,*,0,10,0,*,
      10,10,0,p,0,10,0,*,0,10,10,*
950 END
5400 FOR I=1 TO NN
5410 X(I)=X(I)+XXM:Y(I)=Y(I)+YYM:Z(I)=Z(I)+ZZM
5420 NEXT I
5430 RETURN
```

说明：

15行 将 XM、YM、ZM 置 0，由 3100~3170 子程序使每个立方体移动。图案整体移动由 5400~5430 行子程序完成。

20~70 行 采用二重循环，使立方体在 X 方向和 Y 方向移动，形成并进图形。

程序运行例 1

D	轴测投影
X0,Y0	-50,-50
RT	0.5
XM	0
YM	0
ZM	0
YR,XR,ZR	30,20,0

运行结果如图 5.9 所示。

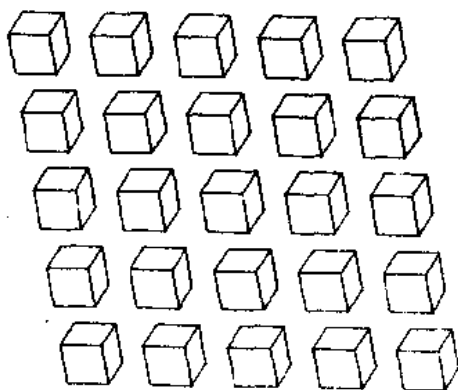


图 5.9

运行结果例 2:

C	三点透视图
X0,Y0	0,0
RT	0.5
XM(=XXM)	-40
YM(=YYM)	-50
ZM	0
SP	200
YR,XR	25,20

运行结果如图 5.10 所示。

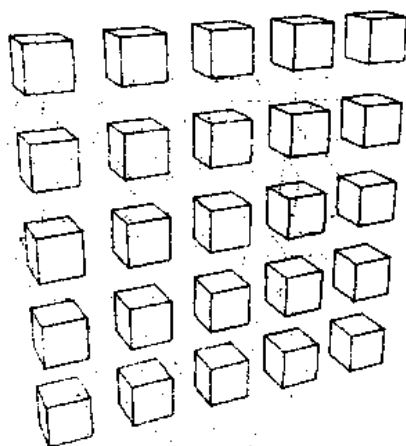


图 5.10

2、大厦透视图

采用并进的方法绘大厦透视图的程序如下。

```
5 'pc5-5(2)
```

```
10 GOSUB 2100: GOSUB 2900: GOSUB 3000: GOSUB 3100
```

```

20 GOSUB 3200: GOSUB 3600: GOSUB 3800
30 DATA 8
40 DATA -50,-70,0,p,1980,-70,0,* ,1980,1150,0,* ,
      -50,1150,0,* ,50,-70,0,* ,50,-70,800,* ,
      -50,1150,800,* ,50,1150,0,*
50 ERASE X,Y,Z,DL$
60 /
70 GOSUB 2900
80 XM=0:YM=0:ZM=0
90 FOR YM=0 TO 1000 STEP 200
100 FOR XM=0 TO 1750 STEP 250
110 RESTORE 160:GOSUB 3000: GOSUB 3100: GOSUB 5400
120 GOSUB 3200: GOSUB 3600: GOSUB 3800
130 NEXT XM
140 NEXT YM
150 DATA 37
160 DATA 0,0,0,p,180,0,0,* ,180,100,0,* ,0,100,0,* ,
      0,0,0,* ,60,100,0,p,60,0,0,* ,120,100,0,p,120,0,0,*
170 DATA 0,0,0,p,0,0,-30,* ,180,0,-30,* ,180,0,0,*
      0,40,0,p,0,40,-30,* ,180,40,-30,* ,180,40,0,*
180 DATA 0,40,-30,p,0,0,-30,* ,20,40,-30,* ,20,0,-30,*
      40,40,-30,p,40,0,-30,* ,60,40,-30,p,60,0,-30,*
      80,40,-30,p,80,0,-30,*
190 DATA 100,40,-30,p,100,0,-30,* ,120,40,-30,p
      120,0,-30,* ,140,40,-30,p,140,0,-30,* ,
      160,40,-30,p,160,0,-30,* ,180,40,-30,p,180,0,-30,*
950 END

```

```

5400 /
5410 FOR I=11 TO NN
5420 X(I)=X(I)+XXM: Y(I)=Y(I)+YYM:
      Z(I)=Z(I)+ZZM
5430 NEXT I
5440 RETURN

```

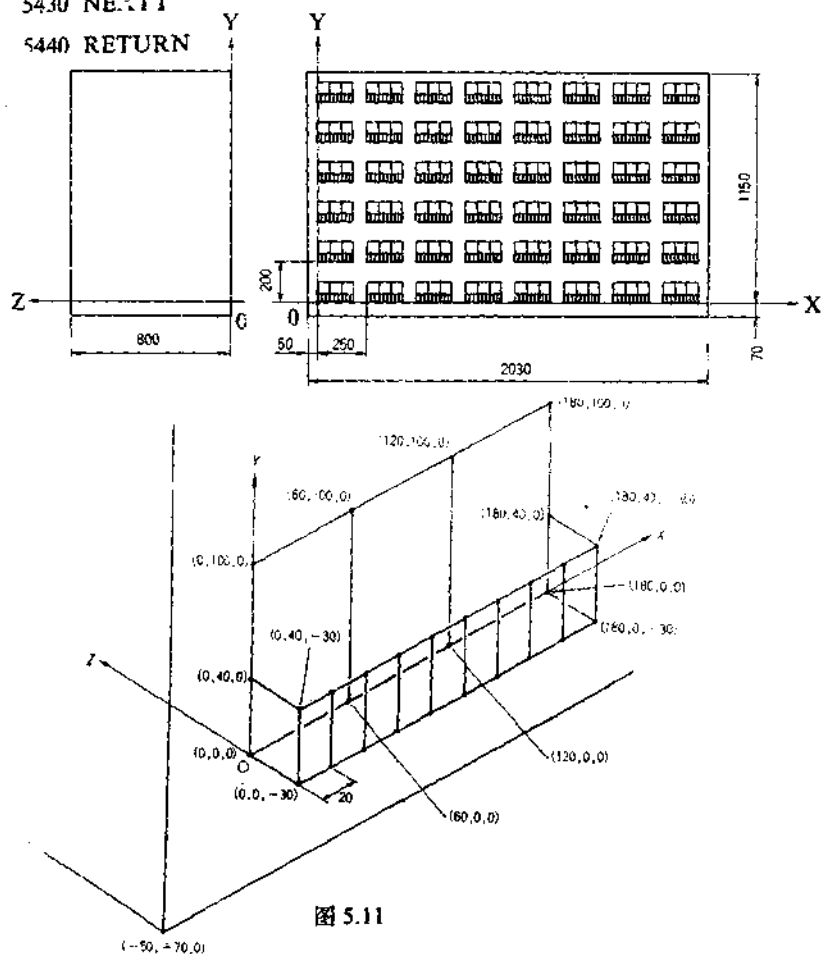


图 5.11

说明:

大厦的数据如图 5.11 所示, 10~40 行为大厦外框的程序; 70~190 行画一组窗框, 然后沿 X 轴、Y 轴并进。

程序运行例 1:

B	二点透视图
X0,Y0	-20,-40
RT	0.1
XM(=XXM)	-400
YM(=YYM)	-180
ZM	0
SP	2500
YR	30

屏幕图形如图 5.12

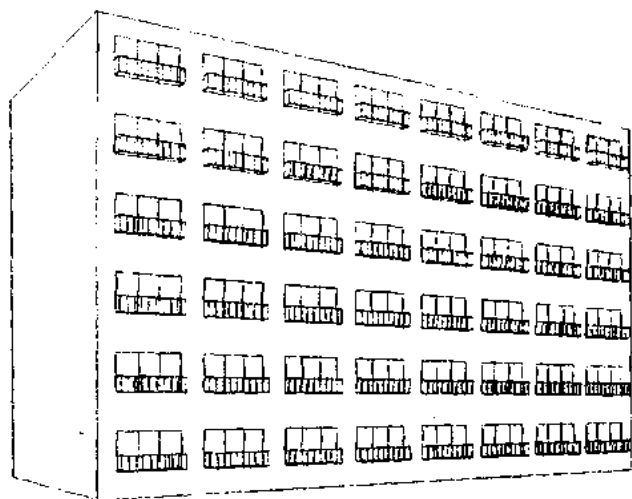


图 5.12

运行例 2:

C	三点透视图
X0,Y0	-20.70
RT	0.1
XM(=XXM)	-350
YM(=YYM)	-1400
ZM	0
SP	2300
YR,XR	25,10

试运行程序，观察图形。

例 5-6 直线图形旋转图案

将直线图形绕 X 轴、Y 轴、Z 轴旋转，可得到各种不同的图案。

1. 长方形绕 X 轴旋转

程序如下:

```

5' pc 5-6(1)
10 GOSUB 2100 : GOSUB 2900
20 FOR SKX=0 TO 2 * PAI STEP PAI / 7
30 RESTORE 110 : GOSUB 3000
40 RESTORE 120 : READ XC, YC, ZC
50 GOSUB 5500 : GOSUB 3100 : GOSUB 3200 : GOSUB 3600
   : GOSUB 3800
60 NEXT SKX
100 DATA 5
110 DATA -10, 20, 0, p, 10, 20, 0, *, 10, 30, 0, *, -10, 30, 0, *,
    -10, 20, 0, *

```

```

120 DATA 0, 0, 0
950 END
5500 FOR I=1 TO NN
5520 Y=Y(I): Z=Z(I)
5530 Y(I)=Z * SIN(SKX)+Y * COS(SKX)+
      YC: Z(I)=Z * COS(SKX)-Y * SIN(SKX)+ZC
5535 X(I)=X(I)+XC
5540 NEXT I
5550 RETURN

```

说明

20~60 行 由 3000~3040 的子程序读入空间长方形顶点坐标数据。由 5500~5550 行子程序求取每旋转 $\pi/7$ 的坐标值，后绘出图形。

120 行为整体图案平移数据。

运行例:

D	轴测投影图
X0, Y0	0, 0
RT	0.5
XM	0,
YM	0
ZM	0
YR, XR, ZR	45, 35.266, 0

屏幕图形如图 5.13。

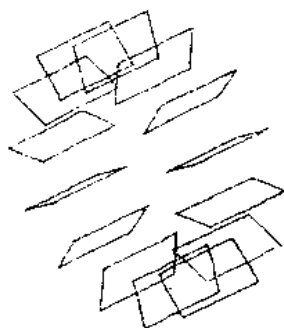


图 5.13

2. 四棱锥体旋转图案

程序如下:

```

5  pc 5-6(2)
10  GOSUB 2100 : GOSUB 2900
20  FOR SKX=0 TO 2*PAI STEP PAI/4
30  RESTORE 110 : GOSUB 3000
40  RESTORE 120 : READ XC, YC, ZC
50  GOSUB 5500 : GOSUB 3100 : GOSUB 3200 : GOSUB 3600
    : GOSUB 3800
60  NEXT SKX
100 DATA 10
110 DATA -10, 30, -10, p, 10, 30, -10, *, 10, 30, 10, *, -10,
    30, 10, *, -10, 30, -10, *, 0, 60, 0, *, -10, 30, 10,
    *, 10, 30, -10, p, 0, 60, 0, *, 10, 30, 10, *
120 DATA 50, 0, 0
950 END
5500

```

```

5510 FOR I=1 TO NN
5520 Y=Y(I): Z=Z(I)
5530 Y(I)=Z * SIN(SKX)+Y * COS(SKX)+YC: Z(I)
      =Z * COS(SKX)-Y * SIN(SKX)+ZC
5535 X(I)=X(I)+XC
5540 NEXT I
5550 RETURN

```

说明

四棱锥体各数据值如图 5.14 所示。程序运行例:

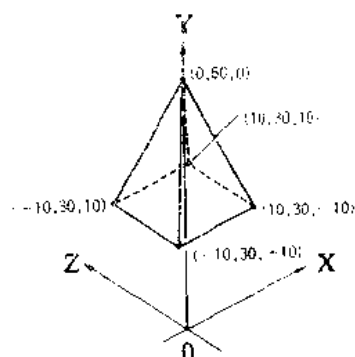


图 5.14

D	轴测投影图
XO, YO	0, 0
RT	0.5
XM	0
YM	0
ZM	0
YR, XR, ZR	45, 35.266, 0

运行结果如图 5.15 所示。

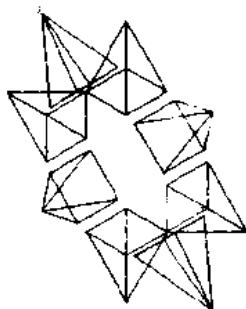


图 5.15

3. 风车扇叶绕 Y 轴旋转

程序如下:

```

5'  pc 5-6(3)
10  GOSUB 2100 : GOSUB 2900
20  FOR SKY=0 TO 2 * PAI STEP PAI / 8
30  -RESTORE 110 : GOSUB 3000
40  RESTORE 120 : READ XC, YC, ZC
50  GOSUB 5600 : GOSUB 3100 : GOSUB 3200 : GOSUB 3600
    : GOSUB 3800
60  NEXT SKY
100 DATA 10
110 DATA 0, -5, 50, p, 0, -1, 50, *, 1, 3, 50, *, 2, 5, 50,
    *, 8, 7, 80, *, 4, 4, 80, *, 1, 1, 80, *, -2, -3, 80,
    *, -4, -7, 80, *, 0, -5, 50, *
120 DATA 0, 0, 0
950 END
5600 /

```

```

5610 FOR I=1 TO NN
5620 X=X(I):Z=Z(I)
5630 X(I)=X * COS(SK Y)-Z * SIN(SK Y):
      Z(I)=X * SIN(SK Y)+Z * COS(SK Y)
5635 X(I)=X(I)+XC:Y(I)=Y(I)+YC:Z(I)=Z(I)+ZC
5636 NEXT I
5638 RETURN

```

说明:

扇叶数据如图 5.16 程序运行例:

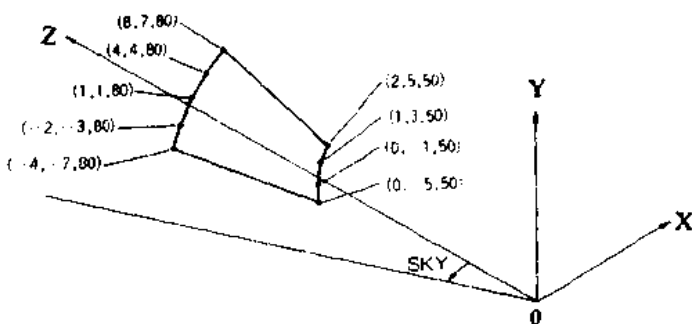


图 5.16

C	三点透视图
X0, Y0	0, 0
RT	0, 7
XM, YM, ZM	0, 0, 0
SP	400
YR, XR	0, 30

运行结果如图 5.17 所示。

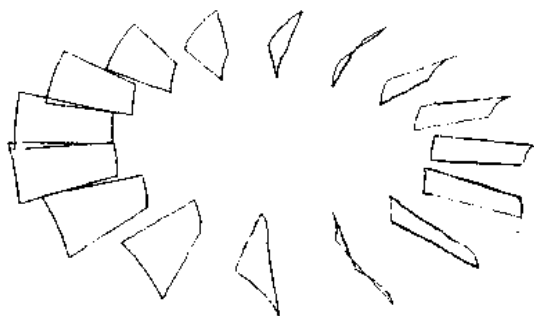


图 5.17

4. 长方体绕 Z 轴旋转图案.

程序如下:

```

5 pc 5-6 (4)
10 GOSUB 2100 : GOSUB 2900
400 FOR SKZ=0 TO 2*PAI STEP PAI/8
410 RESTORE 510 : GOSUB 3000
420 RESTORE 530 : READ XC,YC,ZC
430 GOSUB 5200 : GOSUB 3100 : GOSUB 3200 : GOSUB
    3600 : GOSUB 3800
440 NEXT SKZ
500 DATA 16
510 DATA -5, 50, 0, p, 5, 50, 0, *, 5, 45, 5, *, -5, 45, 5,
    *, -5, 50, 0, *, -5, 70, 20, *, 5, 70, 20, *, 5, 65,
    25, *, -5, 65, 25, *, -5, 70, 20, *, 5, 70, 20, p,
    5, 50, 0, *, 5, 65, 25, p, 5, 45, 5, *
520 DATA -5, 65, 25, p, -5, 45, 5, *
530 DATA 0, 0, 0
    
```



```

550 E=0
5700
5710 FOR I=1 TO NM
5720 X=X(I) : Y=Y(I) : Z=Z(I)
5730 X(I)=X * COS(SKZ)-Y * SIN(SKZ)
      : Y(I)=X * SIN(SKZ)+Y * COS(SKZ)
5735 X(I)=X(I)+XC : Y(I)=Y(I)+YC : Z(I)=Z(I)+ZC
5740 NEXT I
5750 RETURN

```

程序运行例:

C	三点透视图
X0, Y0	0, 0
RT	0.5
XM	0
YM	0
ZM	0
SP	200
VR, ZR	20, 50

运行结果如图 5.13 所示

例 5-7 旋转的飞机图案

程序如下:

```

5 pc 5-7
10 GOSUB 2100 : GOSUB 2900
20 FOR SKY=0 TO 2 * PI-PI/4 STEP PI/4
210 RESTORE 240 : GOSUB 3000 : GOSUB 5800

```

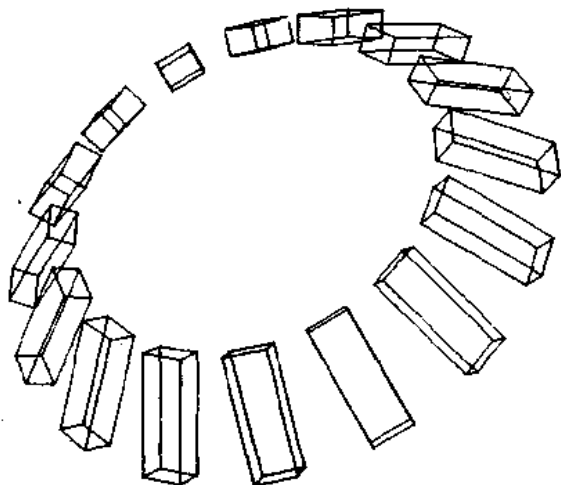


图 5.18

215 RESTORE 320: READ XC, YC, ZC: GOSUB 5770:

GOSUB 3100

220 GOSUB 3200: GOSUB 3600: GOSUB 3800

230 NEXT SKY

235 DATA 53

240 DATA 30, 0, -5, p, 30, -10, -5, *, 30, -10, 5, *, 30, 0, 5,
*, 10, 0, -10, *, -10, 0, -10, *, -45, 0, -5,
*, -45, -5, -5, *, -45, -5, 5, *, -45, 0, 5,
*, -45, 0, -5 *

250 DATA 30, -10, -5, p, 10, -15, -10, *, -10, -15,
-10, -45, -5, -5, *, 30, -10, 5, p, 10, -15, 10,
*, -10, -15, 10, *, -45, -5, 5, *, 30, 0, 5, p, 10,
0, 10, *, -10, 0, 10, *, -45, 0, 5, *

260 DATA 10, 0, -5, p, -10, 0, -5, *, -10, 0, 5, *, 10, 0, 5,

```

      *, 10, 0, -5, *
270 DATA 10, 0, -10, p, 10, 5, -40, *, -5, 5, -40, *, -10, 0,
      -10, *
280 DATA 10, 0, 10, p, 10, 5, 40, *, -5, 5, 40, *, -10, 0, 10,
      *
290 DATA -30, 0, -7, p, -35, 0, -20, *, -45, 0, -20, *, -45,
      0, -5, *
300 DATA -30, 0, 7, p, -35, 0, 20, *, -45, 0, 20, *, -45, 0, 5,
      *
310 DATA -30, 0, 0, p, -35, 15, 0, *, -45, 15, 0, *, -45, 0, 0,
      *, -30, 0, 0, *, 30, 0, 0, p, 30, 0, 0, *, 3, 0, 0, 0,
      *
320 DATA 0, 30, 140
400 ERASE X, Y, Z, DL $
950 END
5770 '
5775 FOR I=1 TO NN
5780 X(I)=X(I)+XC : Y(I)=Y(I)+YC : Z(I)=Z(I)+ZC :
      X=X(I) : Z=Z(I)
5785 X(I)=X * COS(SK Y)-Z * SIN(SK Y)
      : Z(I)=X * SIN(SK Y)+Z * COS(SK Y)
5790 NEXT I
5795 RETURN
5800 '
5805 FOR I=1 TO NN
5810 SXR=40 : SXR=P * SXR
5815 ZZ=Z(I) : YY=Y(I)

```

```

5820 Y(I) = ZZ * SIN(SXR) + YY * COS(SXR)
      : Z(I) = ZZ * COS(SXR) - YY * SIN(SXR)

```

```

5825 NEXT I

```

```

5830 RETURN

```

说明

200~230 读入飞机数据，飞机绕 Y 轴旋转，每 $\pi/4$ 画一架飞机，共画 8 架飞机。飞机尺寸数据如图 5.19 所示。5800~5830 的子程序

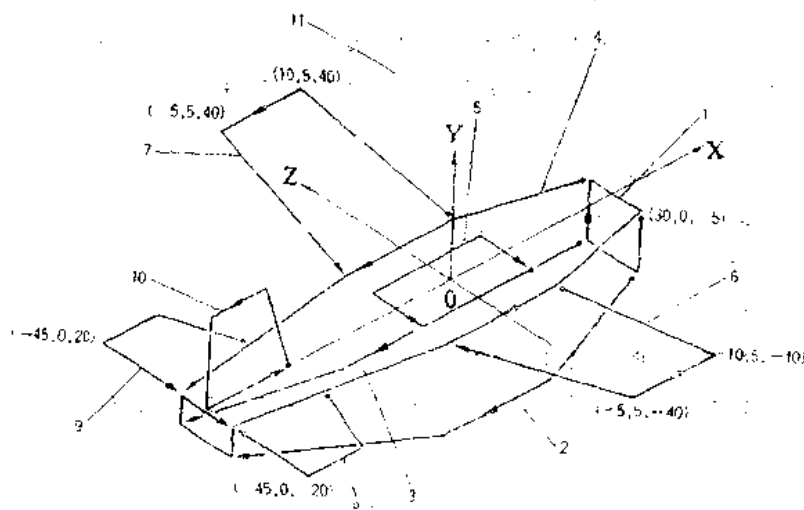


图 5.19

使飞机绕 X 轴旋转 $S \times R$ ，改变 $S \times R$ 的值可使飞机任意倾斜，本程序中取 $S \times R = 40$ 。5770~5790 的子程序给出飞机在 Y 方向的平移量和 Z 方向的平移量。5850~5875 子程序使飞机绕 Y 轴旋转。

程序运行例：

C	3点透视图
X0, Y0	0, 20
RT	0, 5
XM	0
YM	0
ZM	0
SP	600
YR, XR	10, -10

程序运行结果如图 5.20 所示。

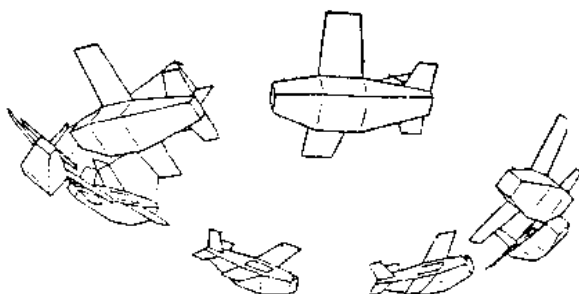


图 5.20

例 5-8 椭圆旋转图案 1

使圆绕 X 轴、Y 轴、Z 轴旋转，可以得到各种各样的投影图案，旋转时图形配置方向不同，所得图案也不同。

1. 星状分布圆环图案

程序如下:

```
5  pc 5-8(1)
10  GOSUB 2100 : GOSUB 2900
20  READ DVN, ZZC
30  GOSUB 4000
40  GOSUB 4100 : GOSUB 5900
100 DATA 80, 10, 40
110 DATA 10, 80, 0, 0, 0, 360
200 XM=XXM : YM=YYM : ZM=ZZM
210 RESTORE 300 : READ DVN, XXC
220 RESTORE 310 : GOSUB 4000
230 GOSUB 4100 : GOSUB 6100
300 DATA 10, -40
310 DATA 10, 0, -80, 0, 0, 360
400 XM=XXM : YM=YYM : ZM=ZZM
410 RESTORE 500 : READ DVN, YYC
420 RESTORE 510 : GOSUB 4000
430 GOSUB 4100 : GOSUB 6300
500 DATA 10, 40
510 DATA 10, 0, 0, 80, 0, 360
950 END
4000 ' read data
4010 READ R, XC, YC, ZC, AS, AE
4020 RETURN
4030 '
4100 '
4110 N=INT(R * RT / (1.5+20)) : BB=2 * PAI / N
```

```

4120 AS=P*AS:AE=P*AE
4130 RETURN
4140 '
5900 '
5910 FOR SXR=0 TO 2*PAI STEP PAI/DVN
5920 FOR I=INT(AS/BB) TO INT(AE/BB)+1
5930 IF I=INT(AS/BB) THEN AA=AS:GOTO 5960
5940 IF I=INT(AE/BB)+1 THEN AA=AE:GOTO 5960
5950 AA=BB*I
5960 ZC2=R*SIN(AA)+ZC:XC2=R*COS(AA)
5970 X(I)=XC2:Y(I)=0:Z(I)=ZC2
5980 Y(I)=Z(I)*SIN(SXR)+YC:Z(I)=Z(I)*COS(SXR)+ZC
      :X(I)=X(I)+XC
5990 GOSUB 3160:GOSUB 3260:GOSUB 3650
6000 IF AA=AS THEN GOSUB 3900:GOTO 6020
6010 GOSUB 3940
6020 NEXT I
6030 NEXT SXR
6040 RETURN
6100 '
6110 FOR SYR=0 TO 2*PAI STEP PAI/DVN
6120 FOR I=INT(AS/BB) TO INT(AE/BB)+1
6130 IF I=INT(AS/BB) THEN AA=AS:GOTO 6160
6140 IF I=INT(AE/BB) THEN AA=AE:GOTO 6160
6150 AA=BB*I
6160 YC2=R*SIN(AA):XC2=R*COS(AA)+XXC
6170 X(I)=XC2:Y(I)=YC2:Z(I)=0

```

```

6180 Z(I)=X(I)*COS(SYR)+ZC:X(I)=X(I)*SIN(SYR)+XC
      :Y(I)=Y(I)+YC
6190 GOSUB 3160:GOSUB 3260:GOSUB 3650
6200 IF AA=AS THEN GOSUB 3900:GOTO 6220
6210 GOSUB 3940
6220 NEXT I
6230 NEXT SYR
6240 RETURN
6250 '
6260 '
6300 '
6310 FOR SZR=0 TO 2*PAI STEP PAI/DVN
6320 FOR I=INT(AS/BB) TO INT(AE/BB)+1
6330 IF I=INT(AS/BB) THEN AA=AS:GOTO 6360
6340 IF I=INT(AE/BB) THEN AA=AE:GOTO 6360
6350 AA=BB*I
6360 YC2=R*SIN(AA)+YYC:ZC2=R*COS(AA)
6370 X(I)=0:Y(I)=YC2:Z(I)=ZC2
6380 X(I)=Y(I)*SIN(SZR)+XC:Y(I)=Y(I)*COS(SZR)+YC
      :Z(I)=Z(I)+ZC
6390 GOSUB 3160:GOSUB 3260:GOSUB 3650
6400 IF AA=AS THEN GOSUB 3900:GOTO 6420
6410 GOSUB 3940
6420 NEXT I
6430 NEXT SZR
6440 RETURN

```

说明:

本程序是使圆分别绕 X 轴、Y 轴、Z 轴旋转，从而绘出星状分布由椭圆构成的圆环图案。

以绕 X 轴旋转的圆为例，由程序的 20 行读入绕 X 轴旋转的圆的个数 DVN 和圆心到 X 轴的距离 ZZC，(如图 5.21(a))，30 行语句读入圆的半径 R、XC、YC、ZC(如图 5.21(b))，圆的起 AS 和终止角 AE。由 5900~6040 的子程序画图。

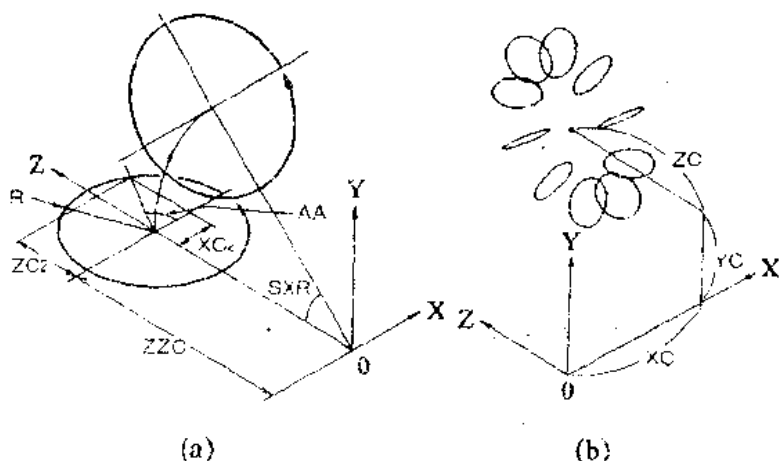


图 5.21

程序运行例:

D	轴测投影
X0, Y0	0, 0
RT	0.3
XM, YM, ZM	0, 0, 0
YR, XR, ZR	30, 20, 0

程序运行结果如图 5.22 所示。

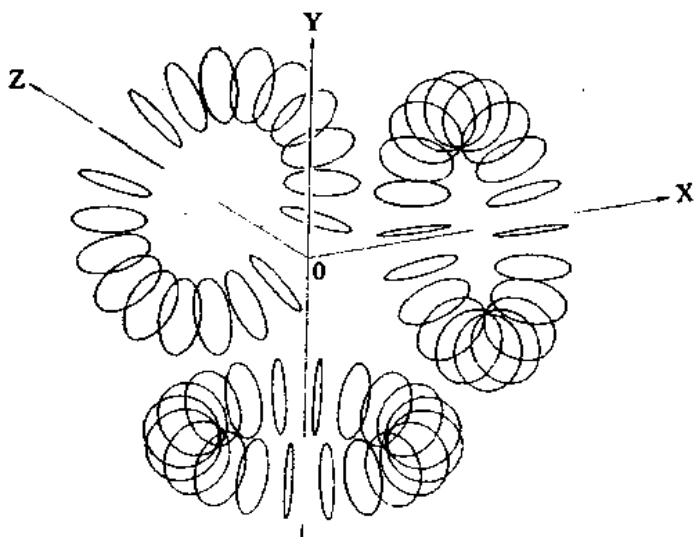


图 5.22

2. 绕 Y 轴旋转的半圆图案程序如下：

5' pc 5-8(2)

10 GOSUB 2100 : GOSUB 2900

210 READ DVN, XXC

220 GOSUB 4000

230 GOSUB 4100 : GOSUB 6100

300 DATA 100, 40, 65

310 DATA 40, 0, 0, 0, 180, 360

950 END

4000' read data

4010 READ R, XC, YC, ZC, AS, AE

4020 RETURN

```

4030 /
4100 /
4110 N=INT(R * RT / 1.5+20) : BB=2 * PAI / N
4120 AS=P * AS : AE=P * AE
4130 RETURN
4140 /
6100 /
6110 FOR SYR=0 TO 2 * PAI STEP PAI / DVN
6120 FOR I=INT (AS / BB) TO INT (AE / BB)+1
6130 IF I=INT (AS / BB) THEN AA=AS : GOTO 6160
6140 IF I=INT (AE / BB) THEN AA=AE : GOTO 6160
6150 AA=BB * I
6160 YC2=R * SIN(AA) : XC2=R * COS(AA)+XXC
6170 X(I)=XC2 : Y(I)=YC2 : Z(I)=0
6180 Z(I)=X(I) * COS(SYR)+ZC : X(I)=X(I) * SIN(SYR)+XC
      : Y(I)=Y(I)+YC
6190 GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6200 IF AA=AS THEN GOSUB 3900 : GOTO 6220
6210 GOSUB 3940
6220 NEXT I
6230 NEXT SYR
6240 RETURN

```

运行例:

C	3点透视图
X0, Y0	0, 0
RT	0.2
XM, YM, ZM	0, 0, 0
SP	400
YR, XR	0, 40

运行结果如图 5.23

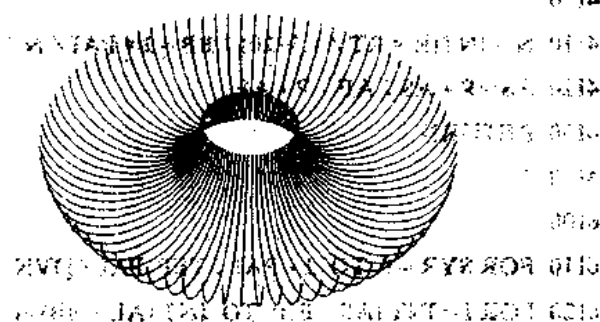


图 5.23

3. 在 X 面上绘 S 图案

D	轴测投影图
X0, Y0	0, 0
RT	0.2
XM, YM, ZM	0, 0, 0
YR, XR, ZR	45, 35.266, 0

按下列参数运行程序

```

5  pc 5-8(3)
10  GOSUB 2100 : GOSUB 2900
20  READ DVN, ZZC
30  GOSUB 4000
40  GOSUB 4100 : PP1=PAI/ 2 : PP2=2* PAI-PA1/ 6 :
    GOSUB 5900
50  RESTORE 120 : READ DVN, ZZC
60  RESTORE 130 : GOSUB 4000
70  GOSUB 4100 : PP1=-PAI/ 2+PAI/ DVN : PP2

```

```

=PAI-PAI / 6 : GOSUB 5900
100 DATA 100, 20, 65
110 DATA 8, 0, -65, 0, 0, 360
120 DATA 20, 65
130 DATA 8, 0, -65, 0, 0, 360
950 END

4000 ' read data
4010 READ R, XC, YC, ZC, AS, AE
4020 RETURN
4030 '
4100 '
4110 N=INT(R * RT / 1.5+20) : BB=2 * PAI / N
4120 AS=P * AS : AE=P * AE
4130 RETURN
4140 '
5900 '
5910 FOR SXR=PP1 TO PP2 STEP PAI / DVN
5920 FOR I=INT (AS / BB) TO INT (AE / BB)+1
5930 IF I=INT (AS / BB) THEN AA=AS : GOTO 5960
5940 IF I=INT (AE / BB)+1 THEN AA=AE : GOTO 5960
5950 AA=BB * I
5960 ZC2=R * SIN(AA)+ZC : XC2=R * COS(AA)
5970 X(I)=XC2 : Y(I)=0 : Z(I)=ZC2
5980 Y(I)=Z(I) * SIN(SXR)+YC : Z(I)=Z(I) * COS(SXR)+ZC
      : X(I)=X(I)+XC
5990 GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6000 IF AA=AS THEN GOSUB 3900 : GOTO 6020

```

6010 GOSUB 3940

6020 NEXT I

6030 NEXT SXR

6040 RETURN

例 5-9 椭圆旋转图案 2

椭圆配置方向不同，旋转后所得图案亦不同。本例中采用不同于上例的配置方向和完全不同的程序，绘出别具一格的图案。

1. 星状分布椭圆图案。

程序如下：

5' pc 5-9 (1)

10 GOSUB 2100 : GOSUB 2900

20 READ DVN, ZZC

30 GOSUB 4000

35 GOSUB 4100 : GOSUB 6500

100 DATA 80, 10, 40

110 DATA 10, 80, 0, 0, 0, 360

200 RESTORE 300 : READ DVN, ZZC

210 RESTORE 310 : GOSUB 4000 : GOSUB 4100 : GOSUB
6700

300 DATA 5, 40

310 DATA 10, 0, -80, 0, 0, 360

320'

400 RESTORE 500 : READ DVN, XXC

410 RESTORE 510 : GOSUB 4000 : GOSUB 4100 : GOSUB
6900

500 DATA 5, 40

```

510 DATA 10, 0, 0, 80, 0, 360
950 END
4000 ' read data
4010 READ R, XC, YC, ZC, AS, AE
4020 RETURN
4030 '
4100 '
4110 N=INT(R * RT / 1.5+20) : BB=2 * PAI / N
4120 AS=P * AS : AE=P * AE
4130 RETURN
4140 '
6500 '
6510 FOR SXR=0 TO 2 * PAI STEP PAI / DVN
6520   FOR I=INT(AS / BB) TO INT(AE / BB)+1
6530     IF I=INT(AS / BB) THEN AA=AS : GOTO 6560
6540     IF I=INT(AE / BB)+1 THEN AA=AE : GOTO
        6560
6550   AA = BB * I
6560   XC2 = R * COS(AA) : YC2 = R * SIN(AA)
6570   X(I)=XC2 : Y(I)=YC2 : Z(I)=ZC : YY = Y(I)
6580   Y(I)=Z(I) * SIN(SXF)+YY * COS(SXF)+YC :
        Z(I)=Z(I) * COS(SXF)-YY * SIN(SXF)+2C :
        X(I)=X(I)+XC
6590 GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6600 IF AA = AS THEN GOSUB 3900 : GOTO 6620
6610 GOSUB 3940
6620 NEXT I

```

```

6630 NEXT SXF
6640 RETURN
6700 '
6710 FOR SYR = 0 TO 2 * PAI STEP PAI / DVN
6720   FOR I = INT(AS / BB) TO INT(AE / BB) + 1
6730     IF I = INT(AS / BB) THEN AA = AS : GOTO 6760
6740     IF I = INT(AE / BB) + 1 THEN AA = AE : GOTO 6760
6750     AA = BB * I
6760     XC2 = R * COS(AA) : YC2 = R * SIN(AA)
6770     X(I) = XC2 : Y(I) = YC2 : Z(I) = ZZC : ZZ = Z(I)
6780     Z(I) = ZZ * COS(SYR) - X(I) * SIN(SYR) + ZC
        : X(I) = ZZ * SIN(SYR) + X(I)
        : * COS(SYR) + XC : Y(I) = Y(I) + YC
6790     GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6800     IF AA = AS THEN GOSUB 3900 : GOTO 6820
6810     GOSUB 3940
6820   NEXT I
6830 NEXT SYR
6840 RETURN
6850 '
6900 '
6910 FOR SZR = 0 TO 2 * PAI STEP PAI / DVN
6920   FOR I = INT(AS / BB) TO INT(AE / BB) + 1
6930     IF I = INT(AS / BB) + 1 THEN AA = AE : GOTO 6960
6940     IF I = INT(AE / BB) + 1 THEN AA = AE : GOTO 6960
6950     AA = BB * I
6960     YC2 = R * SIN(AA) : ZC2 = R * COS(AA)

```



```

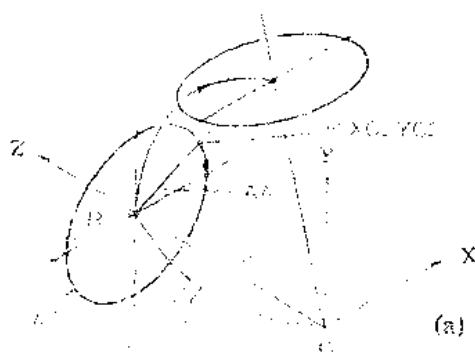
6970  X(I)=XXC:Y(I)=YC2:Z(I)=ZC2:XX=X(I),
6980  X(I)=XX * COS(SZR)-Y(I) * SIN(SZR)+XC:
      Y(I)=XX * SIN(SZR)+Y(I) * COS(SZR)+YC:
      Z(I)=Z(I)+ZC
6990  GOSUB 3160:GOSUB 3260:GOSUB 3650
7000  IF AA=AS THEN GOSUB 3900:GOTO 7020
7010  GOSUB 3940
7020  NEXT I
7030  NEXT SZR
7040  RETURN

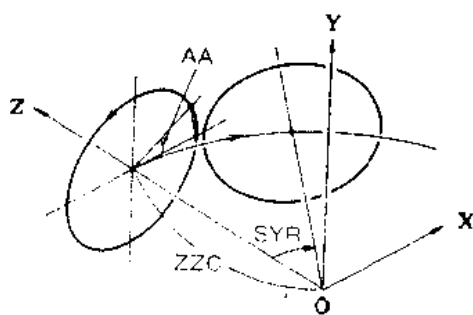
```

说明：本程序中绕各轴旋转的椭圆参数设置如图 5.24(a), (b), (c)所示。

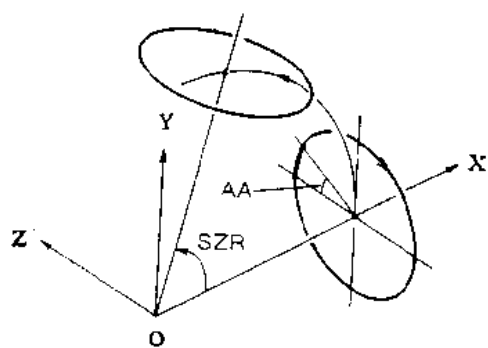
程序运行例：

D	轴测投影
X0, Y0	0, 0
RT	0.3
XM, YM, ZM	0, 0, 0
YR, XR, ZR	30, 30, 0





(b)



(c)

图 5.24

程序运行结果如图 5.25 所示。

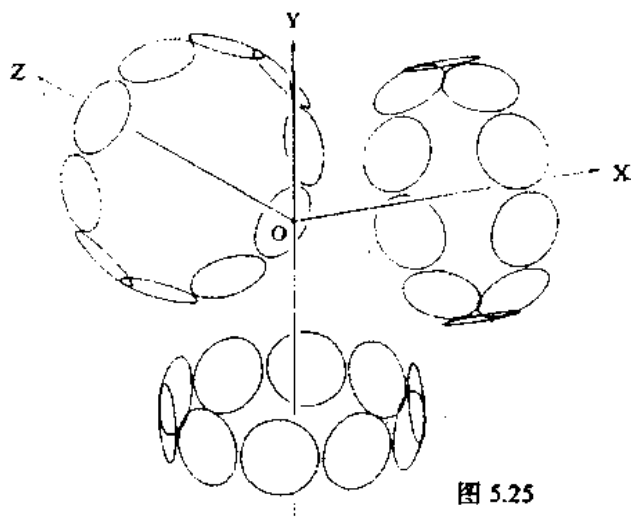


图 5.25

2. 圆环透视图案

程序如下:

```

5' Pc 5-9(2)
10 GOSUB 2100 : GOSUB 2900
15 FOR K=0 TO 10
20 RESTORE 100 : READ DVN, ZZC
30 RESTORE 110 : GOSUB 4000 : XC=XC+30 * K
35 GOSUB 4100 : GOSUB 6500
40 NEXT K
90 DATA 100
100 DATA 10, 65
110 DATA 8, 0, 0, 0, 0, 360
950 END
4000' read data
4010 READ R, XC, YC, ZC, AS, AE

```

```

4020 RETURN
4030 /
4100 /
4110 N=INT(R * RT / 1.5+20) : BB = 2 * PAI / N
4120 AS=P * AS : AE=P * AE
4130 RETURN
6500 /
6510 FOR SXF=0 TO 2 * PAI STEP PAI / DVN
6520   FOR I=INT (AS / BB) TO INT(AE / BB)+1
6530     IF I=INT (AS / BB) THEN AA = AS : GOTO 6560
6540     IF I=INT (AE / BB)+1 THEN AA = AE : GOTO 6560
6550     AA = BB * I
6560     XC2=R * COS(AA) : YC2=R * SIN(AA)
6570     X(I)=XC2 : Y(I)=YC2 : Z(I)=ZC : YY=Y(I)
6580     Y(I)=Z(I) * SIN(SXF)+YY * COS(SXF)+YC :
        Z(I)=Z(I) * COS(SXF)-YY * SIN(SXF)+ZC :
        X(I)=X(I)+XC
6590     GOSUB 3160 : GOSUB 3260 : GOSUB 3650
6600     IF AA=AS THEN GOSUB 3900 : GOTO 6620
6610     GOSUB 3940
6620   NEXT I
6630 NEXT SXF
6640 RETURN

```

程序运行例:

B	2 点透视图
X0, Y0	0, 0
RT	0.5
XM, YM, ZM	0, 0, 0
SP, YR	100, 90

程序运行结果如图 5.26

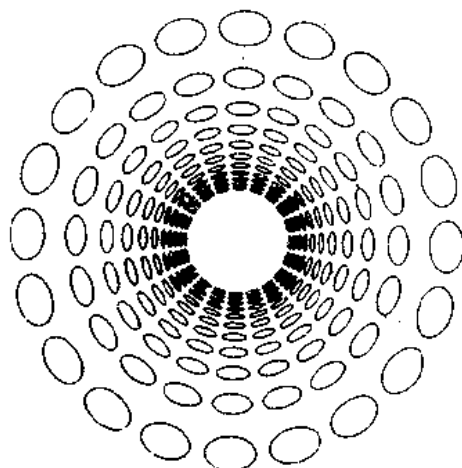


图 5.26

3. 球面状分布的旋转圆图案 i

程序如下:

```

5' pc 5-9(3)
10 GOSUB 2100 : GOSUB 2900
20 INPUT " szr "; SZR : SZR = P * SZR
30 READ DVN
40 GOSUB 4000 : GOSUB 4100 : GOSUB 7100

```

```

100 DATA 100, 4
110 DATA 10, 0, 0, 40, 0, 360
950 END
4000' read data
4010' READ R, XC, YC, ZC, AS, AE
4020 RETURN
4030'
4100'
4110 N=INT(R * RT / 1.5+20) : BB=2 * PAI / N
4120 AS=P * AS : AE=P * AE
4130 RETURN
4140'
7100'
7100 FOR T=0 TO 2 * PAI STEP PAI / DVN
7120     FOR I=INT(AS / BB) TO INT(AE / BB)+1
7130         IF I=INT(AS / BB) THEN AA=AS : GOTO
            7160
7140     IF I=INT(AE / BB)+1 THEN AA=AE : GOTO 7160
7150     AA=BB * I
7160     YC2=R * SIN(AA) : XC2=R * COS(AA)
7170     X(I)=XC2 : Y(I)=YC2 : Z(I)=0
7180     Z(I)=X(I) * COS(SZR)+ZC : X(I)=X(I) * SIN(SZR)+XC :
        Y(I)=Y(I)+YC
7190     ZZ=Z(I) : Z(I)=Z(I) * COS(T)-Y(I) * SIN(T)
7200     Y(I)=ZZ * SIN(T)+Y(I) * COS(T)
7210     GOSUB 3160 : GOSUB 3260 : GOSUB 3650
7220     IF AA=AS THEN GOSUB 3900 : GOTO 7240

```

7230 GOSUB 3940

7240 NEXT I

7250 NEXT T

7260 RETURN

程序运行例:

D	轴测投影图
X0, Y0	0, 0
RT	0, 5
XM, YM, ZM	0, 0, 0
YR, XR, ZR	45, 35.266, 0
SZR	60

程序运行结果如图 5.27

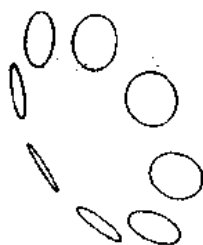


图 5.27

4. 球面分布圆环图案 2

输入下列参数运行程序 PC5-9(4).

C	3点透视图
X0, Y0	0, 20
RT	0.5
XM, YM, ZM	0, 0, 0
SP	200
YR, XR	90, -60
SZR	30

程序运行结果如图 5.28 所示

5' pc 5-9(4)

10 GOSUB 2100 : GOSUB 2900

20 INPUT "szr "; SZR : SZR = P * SZR

30 READ DVN

40 GOSUB 4000 : GOSUB 4100 : GOSUB 7100

100 DATA 100, 10

110 DATA 10, 0, 0, 80, 0, 360

950 END

4000' read data

4010 READ R, XC, YC, ZC, AS, AE

4020 RETURN

4030 /

4100 /

4110 N = INT(R * RT / 1.5 + 20) : BB = 2 * PAI / N

4120 AS = P * AS : AE = P * AE

4130 RETURN

4140 /

7100 /

7110 FOR T = 0 TO 2 * PAI STEP PAI / DVN

7120 FOR I = INT(AS / BB) TO INT(AE / BB) + 1


```

7130 IF I=INT(AS/BB) THEN AA=AS:GOTO 7160
7140 IF I=INT(AE/BB))+1 THEN AA=AE:GOTO 7160
7150 AA=BB*I
7160 YC2=R*SIN(AA):XC2=R*COS(AA)
7170 X(I)=XC2:Y(I)=YC2:Z(I)=0
7180 Z(I)=X(I)*COS(SZR)+ZC:X(I)=X(I)
      *SIN(SZR)+XC:Y(I)=Y(Z)+YC
7190 ZZ=Z(I):Z(I)=Z(I)*COS(T)-Y(I)*SIN(T)
7200 Y(I)=ZZ*SIN(T)+Y(I)*COS(T)
7210 GOSUB 3160:GOSUB 3260:GOSUB 3650
7220 IF AA=AS THEN GOSUB 3900:GOTO 7240
7230 GOSUB 3940
7240 NEXT I
7250 NEXT T
7260 RETURN

```

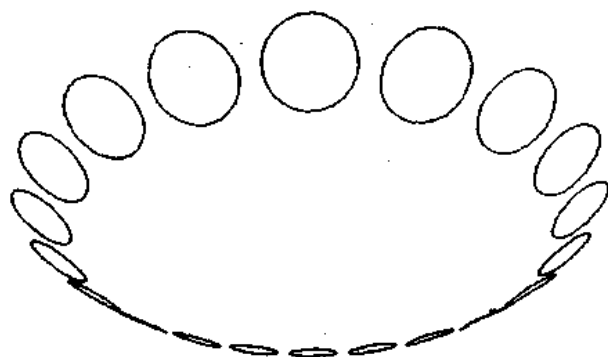


图 5.28

例 5-10 螺线图

1. 竖方向螺线图

程序如下:

```
5' 5-10(1)
10 GOSUB 2100 : GOSUB 2900 : GOSUB 4000 : GOSUB 4100
20 READ MKS, PT : MKS, PT = MKS-1 : GOSUB 7300
100 DATA 300
110 DATA 30, 0, 0, 0, 0, 360
120 DATA 5, 20
950 END
4000' read data
4010 READ R, XC, YC, ZC, AS, AE
4020 RETURN
4030 '
4100 '
4110 N=INT(R * RT / 1.5+20) : BB=2 * PAI / N
4120 AS=P * AS : AE=P * AE
4130 RETURN
4140 '
7300 '
7310 FOR I=INT(AS / BB) TO INT(AE / BB)+1+N * MKS
7320   IF I=INT(AS / BB) THEN AA=AS : GOTO 7350
7330   IF I=INT(AE / BB)+1+N * MKS THEN AA
=AE+2 * PAI * MKS : GOTO 7350
7340   AA=BB * I
7350   ZC2=R * SIN(AA) : XC2=R * COS(AA)
7360   SX2=XC+XC2 : SZ2=ZC+ZC2 : X(I)=SX2 :
```

```

Y(I)=AA/(2*PAI)*PT+YC:Z(I)=SZ2
7370 GOSUB 3160:GOSUB 3260:GOSUB 3650
7380 IF AA=AS THEN GOSUB 3900:GOTO 7400
7390 GOSUB 3940
7400 NEXT I
7410 RETURN

```

输入下列参数运行程序，观察屏幕图形。

D	轴测投影图
X0, Y0	0, 0
RT	0.5
XM, YM, ZM	0, 0, 0
YR, XR, ZR	45, 35.266, 0

2. 螺旋楼梯

应用上述画螺旋图的程序，稍加修改和追加子程序，可以绘螺旋形楼梯。程序如下：

```

5' pc 5-10(2)
10 GOSUB 2100:GOSUB 2900
20 GOSUB 4000:GOSUB 4100:GOSUB 4400
30 RESTOR 130
40 GOSUB 4000:YYC=YC:YC=YC+380:GOSUB 4100:
   GOSUB 4400
50 FOR J=0 TO 2*PAI STEP PAI/5
60 YC=YYC:X1=R*COS(J)+XC:Y(I)=YC:
   Z1=R*SIN(J)+ZC
70 X(I)=X1:Y(I)=Y1:Z(I)=Z1:XX1=X1:YY1=Y1
   :ZZ1=Z1
80 GOSUB 3160:GOSUB 3260:GOSUB 3650:GOSUB 3900

```

```

90 YY1=YC0+360: X(I)=XX1: Y(I)=YY1: Z(I)=ZZ1
100 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB
    3940
110 NEXT J
120 DATA 150
130 DATA 10, 0, -20, 0, 0, 360
200 ERASE X, Y, Z, DL$
210 RESTORE 270: GOSUB 2900
220 FOR SKY=0 TO 2*PAI STEP PAI/8
230 RESTORE 280: GOSUB 3000: GOSUB 7500: GOSUB
    3100
240 GOSUB 3200: GOSUB 3600: GOSUB 3800
250 NN1=NN1+1
260 NEXT SKY
270 DATA 12
280 DATA 10, 0, 0, p, 13, 0, 65, *, -13, 0, 65, *, 0, 0, 10,
    *, 10, -3, 0, p, 13, -3, 65, *, -13, -3, 65, *, 0,
    -3, 10, *, -13, 0, 65, p, -13, -3, 65, *, 13, 0, 65,
    p, 13, -3, 65, *
300 ERASE X, Y, Z DL$
310 RESTORE 120: GOSUB 2900: R=66.3
320 FOR J=PAI/2+2 TO 2*PAI+PAI/2+4 STEP PAI/8
330 X1=R* COS(J): Y1=20NN2: Z1=R* SIN(J):
    XX1=X1: YY1=Y1: ZZ1=Z1
340 X(I)=X1: Y(I)=Y1: Z(I)=Z1
350 GOSUB 3160: GOSUB 3260: GOSUB 3650: GOSUB
    3900

```

```

360  X(I)=XX1 : Y(I)=YY1+50 : Z(I)=ZZ1
370  GOSUB 3160 : GOSUB 3260 : GOSUB 3650 : GOSUB
      3940
380  NN2=NN2+1
390  NEXT J
400  ERASE X, Y, Z, DL $
410  RESTORE 120 : GOSUB 2900 : RESTORE 450 : GOSUB
      4000
420  GOSUB 4100 : RESTORE 500 : READ MKS, PT :
      MKS=MKS-1 : GOSUB 7300
450  DATA 66.3, 0, -39.8, 0, 101, 461
500  DATA 1, 320
950  END
4000'  read data
4010  READ R, XC, YC, ZC, AS, AE
4020  RETURN
4030'
4100'
4110  N=INT (R * RT / 1.5+20) : BB=2 * PAI / N
4120  AS=P * AS : AE=P * AE
4130  RETURN
4140'
4200'
4210  FOR I=INT(AS / BB) TO INT (AE / BB)+1
4220  AA=BB * I
4230  IF I=INT (AS / BB) THEN AA=AS
4240  IF I=INT (AE / BB)+1 THEN AA=AE

```

```

4250 YC2=R * SIN(AA) : ZC2=R * COS(AA)
4260 X(I)=XC : Y(I)=YC+YC2 : Z(I)=ZC+ZC2
4270 GOSUB 3160 : GOUSB 3250 : GOSUB 3650
4280 IF AA=AS THEN GOSUB 3900 : GOTO 4300
4290 GOSUB 3940
4300 NEXT I
4310 RETURN
4320 /
4400 /
4410 FOR I=INT(AS / BB) TO INT (AE / BB)+1
4420 AA=B * I
4430 IF I=INT (AS / BB) THEN AA=AS
4440 IF I=INT (AE / BB)+1 THEN AA=AE
4450 ZC2=R * SIN(AA) : XC2=R * COS(AA)
4460 X(I)=XC+XC2 : Y(I)=YC : Z(I)=ZC+ZC2
4470 GOSUB 3160 : GOSUB 3260 : GOSUB 3650
4480 IF AA=AS THEN GOSUB 3900 : GOTO 4500
4490 GOSUB 3940
4500 NEXT I
4510 RETURN
7300 /
7310 FOR I=INT (AS / BB) TO INT (AE / BB)+I+N * MKS
7320 IF I=INT (AS / BB) THEN AA=AS : GOTO 7350
7330 IF I=INT (AE / BB)+I+N * MKS
      THEN AA=AE+2 * PAI * MKS : GOTO 7350
7340 AA=BB * I
7350 ZC2=R * SIN(AA) : XC2=R * COS(AA)

```

```

7360 SX2=XC+XC2:SZ2=ZC2:X(I)=SX2:
      Y(I)=AA/(2*PAI)*PT+YC:Z(I)=SZ2
7370 GOSUB 3160:GOSUB 3260:GOSUB 3650
7380 IF AA=AS THEN GOSUB 3900:GOTO 7400
7390 GOSUB 3940
7400 NEXT I
7410 RETURN
7500'
7520 FOR I=1 TO NN
7530 X=X(I):Z=Z(I)

```

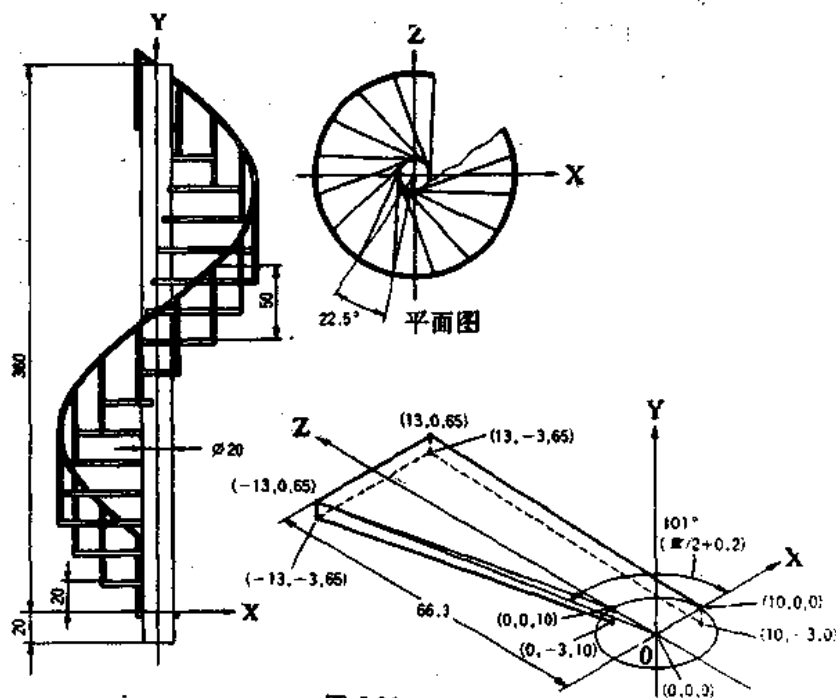


图 5.29

7540 $X(I) = X * \cos(SKY) - Z * \sin(SKY)$

$Z(I) = X * \sin(SKY) + Z * \cos(SKY)$

程序中各参数及数据如图 5.29 所示

7550 $Y(I) = Y(I) + 20 * NN1$

7560 NEXT I

7570 RETURN

程序说明

10~130 行 画中央竖柱;

200~280 行 画楼梯的水色;

300~390 行 画侧面小柱;

程序运行例:

B	2 点透视
X0, Y0	0, -10
RT	0.2
XM	0
YM	-125
ZM	0
SP	350
YR	225

程序运行结果如图 5.30 所示。

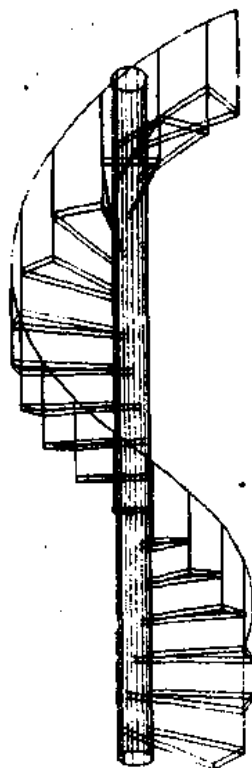


图 5.30

例 5-11 棘轮

5' pc 5-11

```
10 DIM RRMX(50), RRMV(50)
20 GOSUB 2100 : GOSUB 2900 : TN = 20 : R1 = 60 : R2 = 50
30 FOR YDS = 0 TO 20 STEP 20
40 FOR I = 1 TO TN
50 AA = 2 * PAI / TN * I : SZ1 = R2 * SIN(AA) :
    SX1 = R2 * COS(AA)
60 X(I) = SX1 : Y(I) = YDS : Z(I) = SZ1
70 GOSUB 3160 : GOSUB 3260 : GOSUB 3650 : GOSUB 3900
80 SZ2 = R1 * SIN(AA) : SX2 = R1 * COS(AA) : X(I) = SX2 :
    Y(I) = YDS : Z(I) = SZ2
90 GOSUB 3160 : GOSUB 3260 : GOSUB 3650 : GOSUB
    3940
100 IF YDS = 20 THEN 120
110 RRMX(I) = X(I) : RRMV(I) = Y(I) : GOTO 130
120 LINE (RRMX(I), -RRMV(I))-(X(I), -Y(I))
130 AA = 2 * PAI / TN * (I+1)
140 SZ3 = R2 * SIN(AA) : SX3 = R2 * COS(AA) : X(I) = SX3 :
    Y(I) = YDS : Z(I) = SZ3
150 GOSUB 3160 : GOSUB 3260 : GOSUB 3650 : GOSUB
    3940
160 NEXT I
170 NEXT YDS
180 DATA 100
950 END
```

说明:

10 行 定义数组上限
 20 行 输入齿数 TN, 齿尖圆 R_1 、圆底圆 R_2 的值;
 30~170 在 Y 面上画一条齿轮外形轮廓线后取
 YDS=20 再画一次。

40~160 画齿形, 最后完成棘轮图。

程序运行例:

D	轴测投影
X0, Y0	0, 0
RT	0.5
XM, YM, ZM	0, 0, 0
YR, XR, ZR	45, 35.266, 0

程序运行结果如图 5.31 所示。

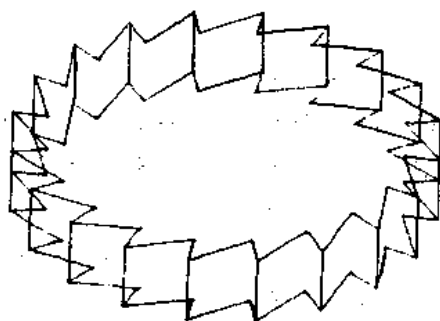


图 5.31

例 5-12 圆盘锯图案

5' pc5-12

10 GOSUB 2100 : GOSUB 2900

20 FOR M=1 TO 2

30 FOR J=0 TO 2*PAI STEP 2*PAI/18

```

40 RESTORE 110 : GOSUB 4000
50 GOSUB 4100 : GOSUB 7600
60 RESTORE 120 : GOSUB 4000
65 IF M = 2 THEN YC = YC - 4
70 GOSUB 4100 : GOSUB 7600
80 NEXT J
90 NEXT M
100 DATA 100
110 DATA 8, 90, 0, 0, 140, 302
120 DATA 30, 66, 0, 2.5, 299, 342
950 END

4000 ' read data
4010 READ R, XC, YC, ZC, AS, AE
4020 RETURN
4030 '
4100 '
4110 N = INT (R * RT / 1.5 + 20) : BB = 2 * PAI / N
4120 AS = P * AS : AE = P * AE
4130 RETURN
4140 '
7600 '
7610 FOR I = INT (AS / BB) TO INT (AE / BB) + 1
7620 AA = BB * I
7630 IF I = INT (AS / BB) THEN AA = AS
7640 IF I = INT (AE / BB) + 1 THEN AA = AE
7650 ZC2 = R * SIN(AA) : XC2 = R * COS(AA)
7660 SX2 = XC + XC2 : SZ2 = ZC + ZC2 : SSX2 = SX2

```

```

7670 SX2= SX2 * COS(J)-SZ2 * SIN(J):
      SZ2= SSX2 * SIN(J)+SZ2 * COS(J)
7680 X(I)= SX2: Y(I)= YC: Z(I)= SZ2
7690 GOSUB 3160: BOSUB 3260: GOSUB 3900: GOTO
      7720
7710 GOSUB 3940
7720 NEXT I
7730 RETURN

```

说明

20~90 行 画厚度为 4 的锯的两个面；其中 30~80 行中转程序 7600 行画齿形。

110~120 行 为绘图数据，120 行中各数据的示意图如图 5.32。

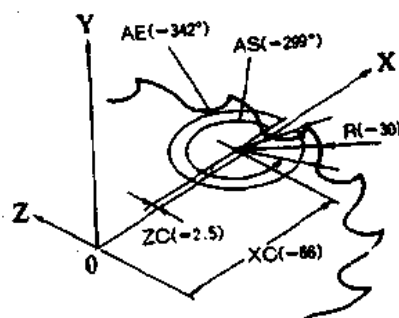


图 5.32

程序运行结果例:

C	3 点透视图
X0, Y0	0,0
RT	0.5
XM, YM, ZM	0,0,0
SP	450
YR, XR	0,40

显示图形如图 5.33 所示。

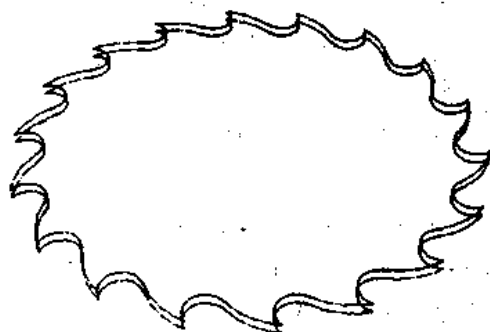


图 5.33

例 5-13 电灯罩

5' pc 5-13

10 GOSUB 2100 : GOSUB 2900

20 DATA 100

100 RESTORE 200 : READ DVN,XXC

110 RESTORE 210 : GOSUB 4000

120 GOSUB 4100 : GOSUB 6100

200 DATA 40,43

210 DATA 5.5,0,-16,0,270,338

220 '

```

300 RESTORE 400 : READ DVN, XXC
310 RESTORE 410 : GOSUB 4000
320 GOSUB 4100 : GOSUB 6100
400 DATA 40, 43
410 DATA 5.5, 0, 16, 0, 22, 90
420 '
500 RESTORE 600 : READ DVN, XXC
510 RESTORE 610 : GOSUB 4000
520 GOSUB 4100 : GOSUB 6100
600 DATA 40, 30
610 DATA 80.3, 0, 0, 0, 167, 193
620 '
700 ERASE X, Y, Z, DL $ : RESTORE 760 : GOSUB 2900
710 FOR SKY=PAI/2 TO 2*PAI+PAI/2 STEP PAI/40
720 RESTOR 770 : GOSUB 3000
730 RESTORE 780 : READ XC, YC, ZC
740 GOSUB 5600 : GOSUB 3100 : GOSUB 3200 : GOSUB
    3600 : GOSUB 3800
750 NEXT SKY
760 DATA 4
770 DATA 30, 21.5, 0, p, 43, 21.5, 0, *, 30, -21.5, 0, p, 43,
    -21.5, 0, *
780 DATA 0, 0, 0
790 '
800 ERASE X, Y, Z DL $ : RESTORE 850 : GOSUB 2900
810 RESTORE 860 : GOSUB 4000 : GOSUB 4100 : GOSUB
    4400

```

```

820 RESTORE 870 : GOSUB 4000 : GOSUB 4100 : GOSUB
    4400
850 DATA 100
860 DATA 30, 0, 21.5, 0, 0, 360
870 DATA 30, 0, -21.5, 0, 0, 360
950 END
4000 ' read data
4010 READ R, XC, YC, ZC, AS, AE
4020 RETURN
4030 '
4110 N=INT(R * RT / 1.5+20) : BB=2 * PAI / N
4120 AS=P * AS : AE=P * AE
4130 RETURN
4140 '
4400 '
4410 FOR I=INT (AS / BB) TO INT (AE / BB)+1
4420 AA=BB * I
4430 IF I=INT (AS / BB) THEN AA=AS
4440 IF I=INT (AE / BB)+1 THEN AA=AE
4450 ZC2=R * SIN(AA) : XC2=R * COS(AA)
4460 X(I)=XC+XC2 : Y(I)=YC : Z(I)=ZC+ZC2
4470 GOSUB 3160 : GOSUB 3260 : GOSUB 3650
4480 IF AA=AS THEN GOSUB 3900 : GOSUB 3900 :
    GOTO 4500
4490 GOSUB 3940
4500 NEXT I
4510 RETURN

```

```

5600'
5610 FOR I=1 TO NN
5620   X=X(I): Z=Z(I)
5630   X(I)=X * COS(SK Y)-Z * SIN(SK Y): Z(I)
       =X * SIN(SK Y)+Z * COS(SK Y)
5640   X(I)=X(I)+XC: Y(I)=Y(I)+YC: Z(I)=Z(I)+ZC
5650 NEXT I
5660 RETURN
6100'
6110 FOR SYR=0 TO 2* PAI STEP PAI / DVN
6120 FOR I=INT (AS / BB) TO INT (AE / BB)+1
6130 IF I=INT (AS / BB) THEN AA=AS: GOTO 6160
6140 IF I=INT (AE / BB) THEN AA=AS: GOTO 6160
6150 AA=BB * I
6160 YC2=R * SIN(AA): XC2=R * COS(AA)+XXC
6170 X(I)=XC2: Y(I)=YC2: Z(I)=0
6180 Z(I)=X(I) * COS(SYR)+ZC: X(I)=X(I) * SIN(SYR)+XC
       : Y(I)=Y(I)+YC
6190 GOSUB 3160: GOSUB 3260: GOSUB 3650
6200 IF AA=AS THEN GOSUB 3900: GOSUB 6220
6210 GOSUB 3940
6220 NEXT I
6230 NEXT SYR
6240 RETRUN

```

说明:

该程序绘制图 5.34 所示数据的电灯罩。程序中

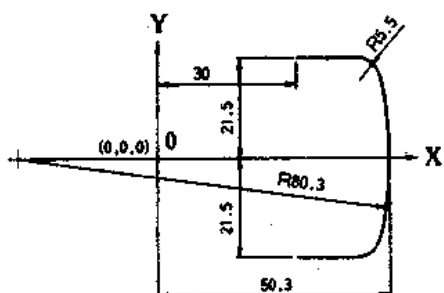


图 5.34

10~210 行，画下部的 R 部分，使用绕 Y 轴旋转的画椭圆子程序 6100~6240，开始角 270° ，终止角 338° ，如图 5.35 所示。

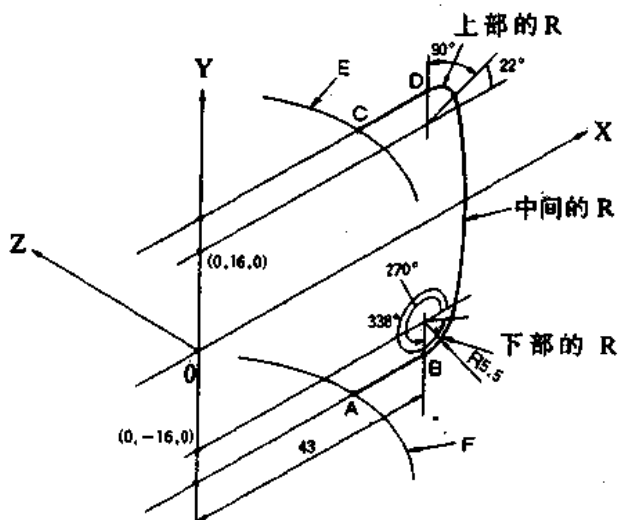


图 5.35

300~410 行 画上部 R 部分。起始角 22° ，终止角

90°.

500~610 行 画中间 R 部分。起始角 167°，终止角 193°，如图 5.36

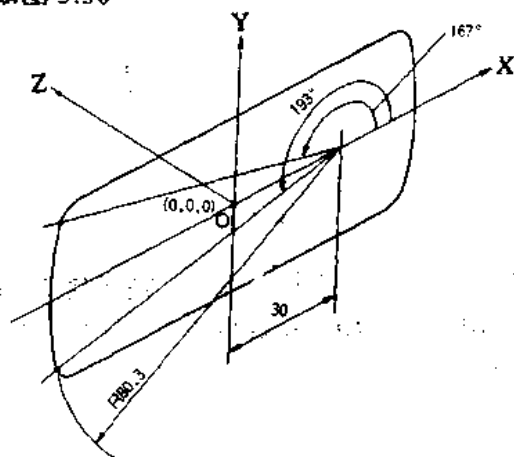


图 5.36

700~780 行 画直线 AB, CD

800~870 行 画圆 E, F. 程序运行例:

C	3 点透视图
X0, Y0	0, 0
RT	0.6
XM, YM, ZM	0, 0, 0
SP	300
YR, XR	0, 45

程序运行后显示图形为图 5.37 所示。

例 5-14 卫星外形图

5 ' PC5-14

10 GOSUB 2100 : GOSUB 2900

20 FOR SKY=0 TO 2 * PAI+PAI / 10 STEP PAI / 10

30 RESTORE 110 : GOSUB 4200 : GOSUB 4300 : GOSUB 3100

```

40  GOSUB 3200 : GOSUB
    3600 : GOSUB 4400
50  NEXT SKY
100 DATA 8
110 DATA 0,0,3,3,3,15,
    25,30,25,40,3,40,3,
    45,0,45
950 END

```

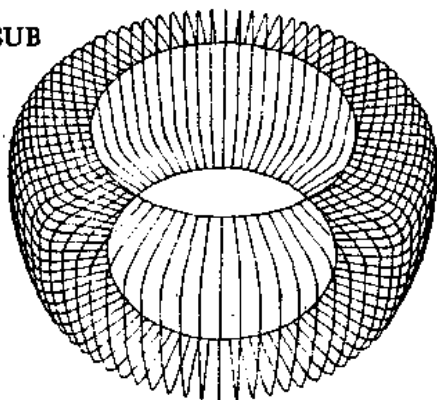


图 5.37

```

4200 FOR I=1 TO NN
4210 READ X(I), Y(I)
4220 NEXT I : RETURN
4230 /
4300 FOR I=0 TO NN
4310 X=X(I) : X(I)=X(I)*COS(SKY) :
    Z(I)=X*SIN(SKY)
4320 NEXT I : RETURN
4400 FOR I=0 TO NN
4410 IF I=1 THEN 4430
4420 LINE (X, -Y)-(X(I), -Y(I))
4430 X=X(I) : Y=Y(I)
4440 IF SKY=0 THEN 4460
4450 LINE (X(I), -Y(I))-(XX(I), -YY(I))
4460 XX(I)=X(I) : YY(I)=Y(I)
4470 NEXT I : RETURN

```

程序数据如图 5.38

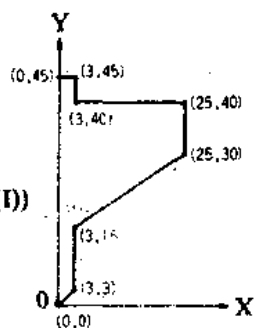


图 5.38

程序运行例:

D	轴测投影图
X0, Y0	0, 0
RT	1
XM, YM, ZM	0, 0, 0
YR, XR, ZR	0, 10, 10

运行结果如图 5.39 所示



图 5.39

例 5-15 汽球

5' pc 5-15

10 GOSUB 2100 : GOSUB 2900

20 FOR SKY=0 TO 2*PAI+PAI/10 STEP PAI/10

30 RESTORE 110 : GOSUB 4200 : GOSUB 4300 : GOSUB
3100

40 GOSUB 3200 : GOSUB 3600 : GOSUB 4400

50 NEXT SKY

100 DATA 20

110 DATA 0, 0, 10, 0, 11, 5, 12, 10, 13, 15, 10, 15, 24.49, 75,
28.72, 80, 31.62, 85, 33.54, 90, 34.64, 95, 35, 100, 34.64, 105,

33.54, 110, 31.62, 115

120 DATA 28.72, 120, 24.49, 125, 18.03, 130, 11.66, 133, 0, 135

950 END

4200 FOR I=1 TO NN

4210 READ X(I), Y(I)

4220 NEXT I: RETURN

4230 '

4300 FOR I=0 TO NN

4310 X=X(I): X(I)=X(I)*COS(SKX): Z(I)=X*SIN(SKX)

4320 NEXT I: RETURN

4400 FOR I=0 TO NN

4410 IF I=1 THEN 4430

4420 LINE (X, -Y)-(X(I), -Y(I))

4430 X=X(I): Y=Y(I)

4440 IF SKX=0 THEN 4460

4450 LINE (X(I), -Y(I))-(XX(I), -YY(I))

4460 XX(I)=X(I): YY(I)=Y(I)

4470 NEXT I: RETURN

利用回转体法画汽球，数据参数如图 5.40.

程序运行例:

D	轴测投影图
X0, Y0	0, 60
RT	0.8
XM, YM, ZM	0, 0, 0
YR, XR, ZR	0, -30, 0

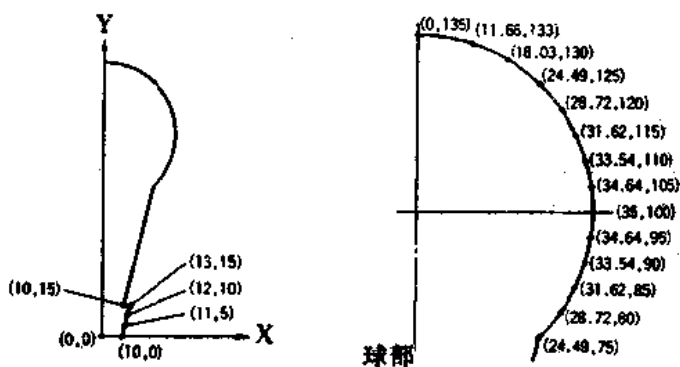


图 5.40

运行结果如图 5.41 所示。

例 5-16 客机

程序如下:

5' pc 5-16

10 GOSUB 2100 : GOSUB 2900

20 FOR SKY=0 TO 2*PAI+PAI/8 STEP/8

30 RESTORE 110 : GOSUB 4200 : GOSUB 4300 : GOSUB
3100

40 GOSUB 3200 : GOSUB 3600 : GOSUB 4400

50 NEXT SKY

100 DATA 14

110 DATA 0, 0, 1, 1, 6, 22, 7.30, 7.5, 40, 7.5, 66, 7.5, 75, 7.5, 88,
7.5, 115, 7, 123, 6, 129, 4, 135, 1, 139.5, 0, 140

210 ERASE X,Y,Z, DL\$,XX,YY : RESTORE 300 : GOSUB
2900

```

220 FOR K=1 TO 2
230 FOR SKY=0 TO
    2*PAI+PAI/5 STEP PAI/5

```

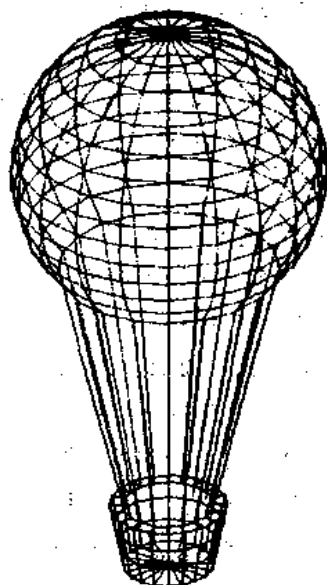


图 5.41

```

240 RESTORE 310 : GOSUB 4200 : GOSUB 4300
250 XCM=24 : YCM=75 : ZCM=6 : IF K=2 : THEN
    XCM=-24
260 GOSUB 5000
270 GOSUB 3200 : GOSUB 3600 : GOSUB 4400
280 NEXT SKY
290 NEXT K
300 DATA 6
310 DATA 2, 0, 2, 8, 5, 3, 8, 4, 8, 4, 8, 15, 4, 19

```

```

400 ERASE X, Y, Z, DL$, XX, YY: RESTORE 500:
    GOSUB 2900
410 RESTORE 510: GOSUB 3000: GOSUB 3100: GOSUB
    3200: GOSUB 3600: GOSUB 3800
500 DATA 29
510 DATA 6.5, 62, 4, p, 22, 60.5, 2, *, 68.5, 43.5, -2.5, *, 68,
    49, -2.5, * -67, 51, -2.5, *, 9, 92, 4, *, 6.5, 95, 4,
    *
520 DATA -6.5, 62, 4, p, -22, 60.5, 2, *, -68.5, 43.5, -2.5,
    *, -68.49, -2.5, *, -67, 51, -2.5, *, -9, 92, 4,
    *, -6.5, 95, 4, *
530 DATA 1.5, 4, -2, p, -27.5, -2, -6, *, 27, 2.5, -6, *, 7,
    19, -2, *, 5.5, 21, -2, *
540 DATA -1.5, 4, -2, p, -27.5, -2, -6, *, -27, 2.5, -6,
    *, -7, 19, -2, *, -5.5, 21, -2, *
550 DATA 0, 8, -3, p, 0, -3, -33, *, 0, 4, -33, *, 0, 27, -10,
    *, 0, 33, -7, *
950 END
4200 FOR I=1 TO NN
4210 READ X(I), Y(I)
4220 NEXT I: RETURN
4230
4300 FOR I=0 TO NN
4310 X=X(I): X(I)=X(I)*COS(SKY)
    : Z(I)=X*SIN(SKY)
4320 NEXT I: RETURN
4400 FOR I=0 TO NN

```



```

4410 IF I=1 THEN 4430
4420 LINE (X,-Y)-(X(I),-Y(I))
4430 X=X(I) : Y=Y(I)
4440 IF SKY=0 THEN 4460
4450 LINE (X(I),-Y(I))-(XX(I),-YY(I))
4460 XX(I)=X(I) : YY(I)=Y(I)
4470 NEXT I : RETURN
5000 '
5100 FOR I=1 TO NN
5020 X(I)=X(I)+XM : Y(I)=Y(I)+YM
      : X(I)=Z(I)+ZM
5030 X(I)=X(I)+XMM : Y(I)=Y(I)+YMM
      : Z(I)=Z(I)+ZMM
5040 NEXT I
5050 RETURN

```

采用尺寸数据如图 5.42 所示。

程序运行例:

D	轴测投影图
X0, Y0	0, 30
RT	1
XM, YM, ZM	0, 0, 0
YR, XR zR	200, 120, 30

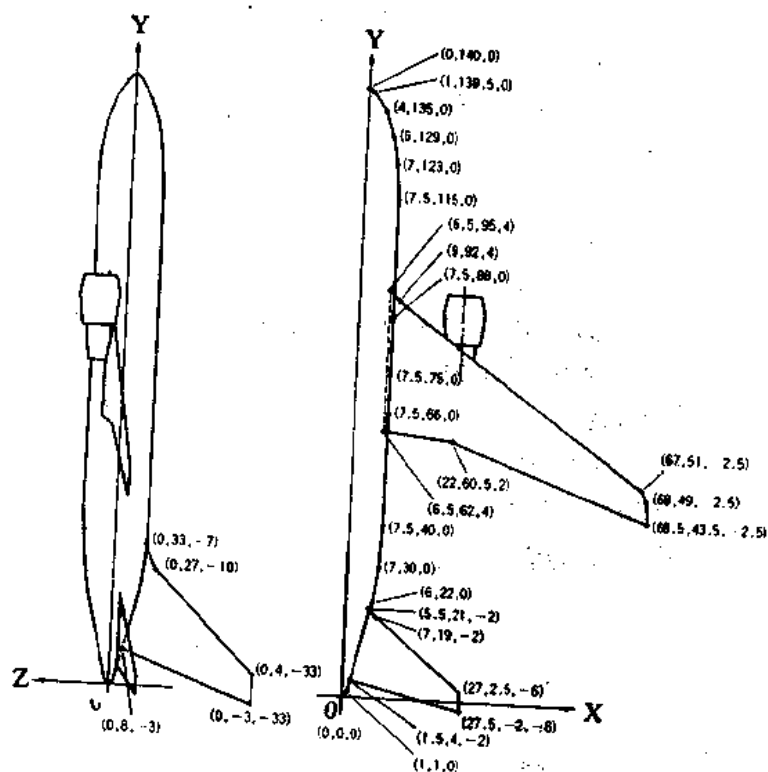


图 5.42

运行结果如图 5.43 所示。

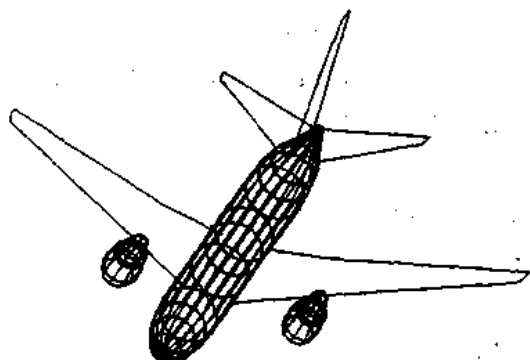


图 5.43

例 5-17 花瓶

利用旋转曲线画花瓶

5' pc5-17

```

10 GOSUB 2100:NN=200
20 DIM XX(NN),YY(NN),A(NN),B(NN),M1(NN),M2(NN)
30 DIM T(NN),M(NN),Y1(NN),X1(NN),X(NN),Y(NN),Z(NN)
40 N=0:ST=5
50 INPUT "n=?"
60 INPUT N
100 FOR I=0 TO N-1
110 INPUT "x";XX(I)
120 INPUT "y";YY(I)
130 NEXT I
140 XX(N)-XX(N-1):YY(N)=YY(N-1)
150 FOR SKY=0 TO 2*PAI STEP 2*PAI/30

```

```

160 X(I)=XX(0):Y(I)=YY(0):Z(I)=0
170 X=X(I):X(I)=X(I)*COS(SK*Y):Z(I)=X*SIN(SK*Y)
180 GOSUB 3160:GOSUB 3260:GOSUB 3650
190 PSET (X(I),-Y(I))
200 GOSUB 1000
210 NEXT SK*Y
950 END
1000'
1010 T(0)=0
1020 FOR I=1 TO N-1
1030 T(I)=T(I-1)+SQR(((XX(I)-XX(I-1))2
      2)+(YY(I)-YY(I-1))2)
1040 NEXT I
1050 FOR FL=1 TO 2
1060 FOR P=0 TO N-1
1070 IF FL=1 THEN Y(P)=XX(P) ELSE Y(P)=YY(P)
1080 X(P)=T(P)
1090 NEXT P
1100 AA=2:P=1
1110 F=(Y(P+1)-Y(P))/(X(P+1)-X(P))
1120 G=(Y(P)-Y(P-1))/(X(P)-X(P-1))
1130 L=(X(P+1)-X(P))/(X(P+1)-X(P-1))
1140 BB=6*(F-G)/(X(P+1)-X(P-1))
1150 A(P)=AA:B(P)=BB
1160 FOR P=2 TO N-2
1170 E=(X(P)-X(P-1))/(X(P+1)-X(P-1))
1180 F=(Y(P+1)-Y(P))/(X(P+1)-X(P))

```

```

1190 G=(Y(P)-Y(P-1))/(X(P)-X(P-1))
1200 D=6*(F-G)/(X(P+1)-X(P-1))
1210 BB=D-E*BB/AA:AA=2-E*L/AA:A(P)=AA:B(P)
      =BB
1220 NEXT P
1230 P=N-2:M(P+1)=0:Z=B(P)/A(P):M(P)=Z
1240 FOR P=N-3 TO 1 STEP-1
1250 L=(X(P+1)-X(P))/(X(P+1)-X(P-1)):Z=(B(P)-L*Z)/
      A(P)
1260 M(P)=Z
1270 NEXT P
1280 M(0)=0
1290 FOR I=0 TO N-1
1300 IF FL=1 THEN M1(I)=M(I) ELSE M2(I)=M(I)
1310 NEXT I
1320 NEXT FL
1330 T=T(0)
1340 J=0
1350 J=J+1:IF J>N THEN GOTO 1450
1360 IF T>=T(J) THEN GOTO 1350
1370 H=T(J)-T(J-1):U=T(J)-T:V=T-T(J-1)
1380 X=M1(J-1)*((U*3)/H-U*H)/6+M1(J)*((V*
      3)/H-V*H)/6+U*XX(J-1)/H+V*YY(J)/H
1390 Y=M2(J-1)*((U*3)/H-U*H)/6+M2(J)*((V*
      3)/H-V*H)/6+U*YY(J-1)/H+V*YY(J)/H
1400 X(I)=X:Y(I)=Y:Z=0
1410 X=X(I):X(I)=X(I)*COS(SKY):Z(I)=X*SIN(SKY)

```

1420 GOSUB 3160:GOSUB 3260:GOSUB 3650

1430 LINE-(X(I),-Y(I))

1440 T=T+ST:GOTO 1340

1450 RETURN

程序运行例(输入数据如图 5.44)

C	2点透视图
X0,Y0	0,-25
RT	0.5
XM,YM,ZM	0,0,0
SP	200
YR,XR	0,30

程序运行结果将如图 5.45 所示。

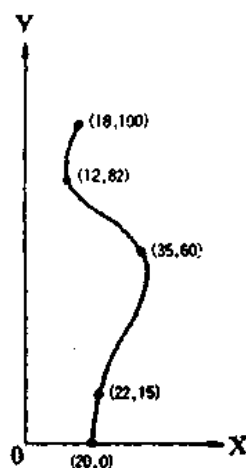


图 5.44

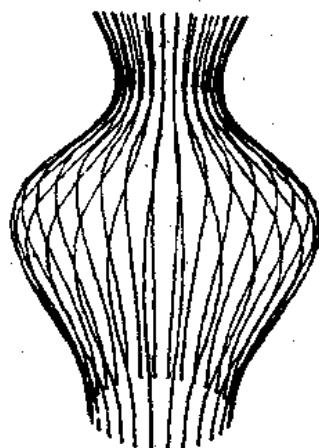


图 5.45

附录 IBM PC BASIC 绘图辅助教学软件(BG1.0)

为帮助初学计算机绘图的读者，本书向大家推荐《IBM PC BASIC 绘图辅助教学软件》(BG1.0) 版，需要的读者请参阅本书后封的说明。

本软件主要介绍高级 BASIC 语言 2.0 版的绘图功能。软件的源程序全部用高级 BASIC 语言 2.0 版编写，软件简称 BG (BASIC GRAPHICS)

一、BG 的功能

1. BG 为初级用户提供了一个打开计算机绘图之门的钥匙。

(1) BG 配有一个用 BASIC 语言顺序文件制作的高级 BASIC 语言 2.0 版的信息库，库中存有 BASICA 2.0 版的全部 175 条语句，命令和函数的信息。包括语句（命令、函数）的名称、语法格式、例程和简要的中文说明。用户利用 BG 中汇编语言编写的动态窗口快速查询方式，查询自己所要了解的某一条语句（命令、函数）的信息。

(2) 能使初级用户直观、详尽地了解 BASICA 2.0 版的绘图功能。

BG 在每一条绘图语句的后面都附有一个语句解释程序，这些解释程序针对不同语句的特点，或者采取将该语句（命令、函数）的所有功能演示一遍的方法（如 POINT 函数，GET 语句、PUT 语句等），或者采用引导用户自己动手操作，改变各种参数的方法（如 DRAW 语句，COLOR 语句）等，以加深对语句（命令、函数）功能的认识。

(3) BG 给出九类 BASIC 语言绘图的应用程序。诸如动画制作、统计图制作、计算机游戏、电脑辅助教学、计算机音乐等。

2. 对于中、高级用户 BG 不但是一本简洁、实用的语句、命令、函数的速查手册，而且还是一本具有大量生动、形象实例的提高 BASIC 语言绘图程序设计技巧的入门工具。

2. BG 的运行环境

(1) IBM PC / XT 及其兼容机

(2) 配有 CGA 卡

(3) 内存容量 > 128KB

3. BG 销售盘上的文件。

Volume in drive A has no label

Directory of A: ¥

BASICA	COM	26112	10-20-83	12:00p
MAINMENU	BAS	1443	1-01-80	1:30a
DEMOMENU	BAS	1991	1-01-80	1:09a
BASMENU	BAS	5598	1-01-80	1:30a
INSTRLIB	DAT	61425	6-29-89	11:04a
GRAPHIC	DAT	16392	4-19-89	6:10p
PAGE1	DAT	4104	1-02-80	12:03a
BLOAD	DAT	16392	1-02-80	3:06a
INNER	BAS	923	4-13-89	9:50p
CAR	BAS	2038	1-01-80	12:43a
MULTIPLY	BAS	3786	1-01-80	7:01p
SPACE	BAS	985	1-01-80	12:37p
BUTTER1	BAS	1712	1-01-80	7:10p
TONGGITU	BAS	1732	1-01-80	7:05p

MANYWIN	BAS	2200	1-01-80	1:10a
MUSICART	BAS	5484	1-01-80	12:25a
LINCHANG	BAS	1303	1-01-80	7:13p
BEE	BAS	4985	1-01-80	12:32a
ASC	BAS	606	4-19-89	3:39p
CIRCLE	BAS	658	1-01-80	12:58a
COLOR	BAS	3790	1-01-80	12:57a
BSAVE	BAS	821	4-19-89	3:53p
DRAW	BAS	1265	4-19-89	4:28p
GET	BAS	431	4-19-89	6:36p
LINE	BAS	434	4-19-89	4:00p
PAINT	BAS	1787	4-19-89	7:38p
POINT	BAS	505	4-19-89	4:07p
PSET	BAS	228	4-19-89	7:41p
SCREEN	BAS	1356	4-19-89	5:02p
STRING \$	BAS	661	4-19-89	4:47p
VIEW	BAS	685	4-19-89	4:57p
CHR \$	BAS	715	4-19-89	4:22p
BOOK	BAS	2159	1-01-80	12:59a
P	BAS	2	1-01-80	12:03a
LBOOK2	BAS	2717	1-01-80	2:47a
VERSION	BAS	636	1-01-80	12:41a
DANCE	BAS	9714	1-01-80	1:09a
CAT1	BAS	1236	1-01-80	12:20a
BG	BAT	29	1-01-80	2:17a
BASCOV	BAS	510	1-01-80	1:30a
TAIKONG		6940	4-19-89	4:25p

41 File(s)

102400 bytes free

下面把其中主要文件介绍一下

MAINMENU.BAS: 主控模块菜单文件。

BASCOUER.BAS: 分控模块 1 (基本语句部分) 入口文件
BASMENU.BAS: 分控模块 1 (基本语句部分) 主文件
DEMOMENU.BAS: 分控模块 2 (演示程序部分) 菜单文件
INSTRLIB.DAT: 基本语句信息库

下面是所有的解释程序文件(共 14 个)

ASC.BAS CIRCLE.BAS COLOR.BAS
BSAVE.BAS DRAW.BAS CHR\$.BAS
GET.BAS LINE.BAS PAINT.BAS
POINT.BAS PSET.BAS SCREEN.BAS
STRING\$.BAS VIEW.BAS

以下是九类演示程序

TONGGITU.BAS 统计用图
BUTTER1.BAS }
LINCHANG.BAS } 三维及艺术图形

DANCE.BAS 舞蹈家

MVLTIPLY.BAS CAI 辅助教学

CAR.BAS 公路赛车

MUSICART.BAS 计算机音乐艺术

MANYWIN.BAS }
INNER.BAS } 窗口技术

BEE.BAS 动画——蜜蜂采蜜

SPACE.BAS 太空飞船

如果在 BASIC 状态下运行 BOOK.BAS, 可以在屏幕上得到一些帮助信息。

如果把打印机接好, 并且能够打印中文的话, 那么在 OK 状态下敲入 RUN "LBOOK2" 那么您将得到一份打印好

的帮助信息。

另外，为了限制那些通过非法途径获得 BG 的人，我们把一些文件进行了加密存贮。如果您在调入某一个文件后，想列它的清单时可能会出现：

Illegal function call

这样的出错信息。没关系，只要您键入：

LOAD "P.BAS"，然后再键入 LIST 就可以了。

二、BG 的设计思路

BG 采用模块化设计：对于无参数传递的模块之间，以 RUN“文件名”的方式进行连接；对于需要进行参数传递的模块之间，则用 CHAIN 行号”的方式进行连接。

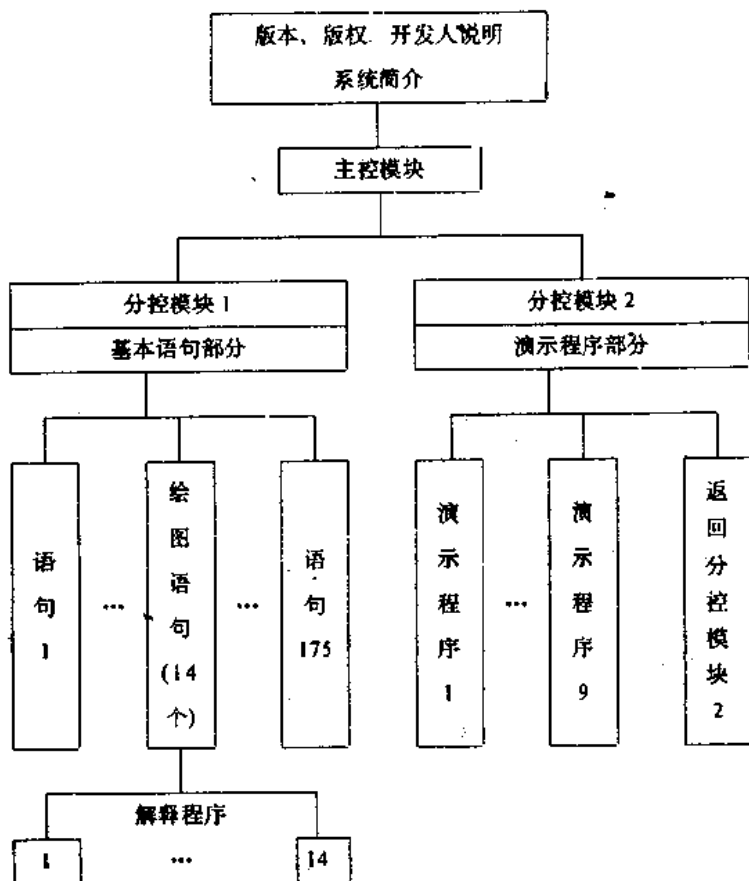
BG 的主框图见附图 1。

1. 屏幕布局说明

为了美观、简洁，我们把所有 175 条语句（命令、函数）的屏幕设计成相同的格式，下面以 CHR\$ 函数为例进行说明：

语句： CHR\$	说 明
格式：V\$ = CHR\$ (N)	把一个 ASCII 代码转换成它相应的字符。N 取值从 0 到 255。利用此命令除了可以产生标准字母与数字外，还能产生许多图形符号，更能产生有用的图形。
按→进入演示程序。 按<ESC>键返回菜单	例如：PRINT CHR\$ (1) 将在屏幕上显示一个笑脸

屏幕为左右两部分。左边第一栏中为语句（命令，函数）的名称。第二栏为格式栏，如果该语句（命令，函数）



附图 1

有多种格式，则将其全部列出。格式说明方法和原 IBM PC BASIC 的格式说明方法相同，即中括号 [] 中为可省略项。第三栏为信息提示栏，共两行，第一行注明此语句（命令，函数）后面是否有解释程序，当无语句解释程序时此行为空白；第二行提示热键 <ESC> 键返回上一级分控菜单，屏幕

的右边为该条语句（命令、函数）码简要中文注释和一个简单的应用举例。（注：在应用举例中所有的单引号在实际应用中应改为双引号）。用户看过简介，再实际将实例操作一遍后，便可对该语句（命令、函数）有一个初步的了解，掌握它的初步应用。

2. 绘图语句解释程序

绘图语句功能介绍是 BG 的主要内容之一，为使初级用户理解方便，针对不同语句（命令、函数）的特点，我们采用了不同的解释方法，下面分别举例：

(1) 参数自动变化，动能自动演示。以 GET 语句和 PUT 语句为例，这是一个互逆过程。功能分别为：将屏幕上用户设定大小的矩形区域内的图形存入一个指定数组和从一个指定数组中将其放回到屏幕上用户任意指定的位置。执行这两条语句的解释程序时，用户会看到一个图形在瞬间被复制到整个屏幕内。

(2) 由用户自行操作改变参数的值。以 DRAW 语句为例：DRAW 语句后面跟着多种字符串子命令来实现多种绘图功能，相当于 BASIC 语言中的绘图子语言。由于子命令多而复杂，用户难以从几个自动演示例子中弄懂各种子命令的具体功能，因此我们采用了下面的方法：见图 3~图 5。当用户键入一串命令字符串后，解释程序自动将其产生的实际效果显示在屏幕上，并将该命令串显示在的画图形的上方。

语句: DRAN	说明 千万别忽略这个语句，DRAN是 IBM PC BASIC 与其它 BASIC 的重
格式: DRAN 字符串	

	要区别。它可称为绘图子语言， DRAN 按照字符串的说明画一个目 标图形。本语句的解释程序将使你 很容易地实践一下各种不同的 DRAN 命令。
按→进入解释程序	例如: 5 SCREEN 1: A=20 10 DRAN/U=A;R=a;D=a;L=a; 20 DRAN/E15 F15 L30'
按<ESC>键返回菜单	

—————DRAN 解释—————

"Mx, y"绝对位置移动

"M±x, y"相对位置移动

HUE

"B(子命令)"移动, 但不画出任何点

L * R

"N(子命令)"移动, 完成后返回初始位置

GDF

"An"角度(0-3, 缺省 0)

"Pp,b" 颜色(0-3, 缺省 3)

"Sa"比例(1-255), 缺省 4)

"TAn" 旋转(-360 度到+360 度)

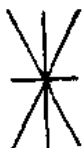
"Pp,b"着色(p:颜色, b:边界)

"x (字符串)"执行命令

"=(变量)"变量标识符

输入n个命令字符串...nu33 ne33 or33 nf33 nd33 ng33 nl33 nb33

all33 ae33 ar33 af33 ad33 ad33 ag33 al33 ab33



按任一健继续...

按<ESC>键退出

附图 2

下面分别说明各条绘图语句（函数）的解释程序：

(1) 函数 ASC()：此函数可以使用户得到计算机中使用的任一字符的 ASCII 码值，当进入这个函数的解释程序后，在屏幕提示下，键入任一键盘所能键入的字符，均能得到其字符相对应的 ASCII 码值（有些 ASCII 码对应的是控制命令，不能用于这个解释程序）。当键入 <ESC> 键时，返回到一级菜单。

(2) 语句 BSAVE 和 BLOAD：BSAVE 语句用来存贮一幅整个屏幕的画面，BLOAD 语句用来将这幅画面显示在屏幕上。运行这两个语句的解释程序时，屏幕上首先画出一幅图案（快乐的小孩脸），并着有颜色，然后程序利用 BSAVE 语句将其自动存贮起来，擦净整个屏幕，再将存贮的图案用 BLOAD 语句显示出来，从中你可以了解到这两个语句使用后产生的效果及运行速度等问题。

(3) 函数 CHR\$()：此函数与 ASC() 函数是一对互逆的过程。它可以得到任一 ASCII 码值所对应的字符。

(4) 语句 CIRCLE：此语句用来画圆、圆弧、椭圆、扇形等。CIRCLE 语句后面跟参数较多，其语句格式为：

CIRCLE (X 坐标, Y 坐标), 半径, 颜色值, 起始角度, 终止角度, 纵横轴比例

当进入解释程序，通过键入以上列出的各种参数值后，即可了解 CIRCLE 语句在不同参数下所产生的效果。

(5) 语句 COLOR：

①在图形方式下的 COLOR 语句用来设定背景颜色和所采用的调色板号。进入解释程序后，键入一次 <F1>

键，即使背景颜色值加 1；键入一次 <F2> 键，使调色板号加 1。其实质是通过改变 COLOR 语句的两个参数值，得到 COLOR 语句的所有组合形式，使您看清每种组合所产生的屏幕效果；并给出颜色的中文说明。

②在文本方式下 COLOR 语句用来设定前景颜色，背景颜色和边框颜色。

进入解释程序后，屏幕上提示：

<F1>.....改变前景颜色
<F2>.....改变背景颜色
<F3>.....改变边框颜色
<F4>.....退出

键入 <F1>、<F2>、<F3> 将分别使前景色，背景色，边框色加 1。其实质是改变 COLOR 语句的三个参数值，得到 COLOR 语句在文本方式下的所有组合形式的屏幕效果。COLOR 语句的用法很多，通过认真摸索，会得到炫丽多彩的画面。

③DRAW(略)见前述

④语句 GET 和 PUT(略)见前述

⑤语句 LINE：此语句可以用来画线，画矩形和画方块。线和和方块是绘图中较常用的，这个解释程序展示了 LINE 语句的这三种用法。

⑥语句 PAINT：此语句将一定范围内的屏幕着上指定

颜色。一种方法是以点涂满整个着色区域；另一种方法是用一种由用户设计的花纹涂满整个着色区域，一般称之为“盖瓦”技术。

⑦“盖瓦”技术是 BASICA 2.0 版所特有的，这个解释程序主要介绍 PAINT 语句的这种功能。

当进入解释程序后，程序询问采用何种绘图方式：中分辨率或高分辨率。若进入中分辨率方式下，则程序要求用户较入背景颜色和调色板号，这是 PAINT 语句的参数。然后，程序要求用户输入“盖瓦”字节号和“盖瓦”字节，若在高分辨率方式下，则直接要求用户输入“盖瓦”字节号和字节。之后，程序将绘制一幅画面，并用所设定的盖瓦字节涂满整个画面，使您能明确地了解“盖瓦”的作用。

⑧函数 POINT：此语句得到屏幕上某一点的颜色值。进入解释程序后，屏幕上有一光环自上而下移动，每次取得光环中心所经过点的 POINT 函数值，并且显示在屏幕上。

函数 POINT 可以应用于函数放大。BG 中出现重叠画面“graphic”就是利用函数 POINT 将标准 5×7 点阵字符串“graphic”放大绘制出来的。当然，用 POINT 函数还可将字符翻转，斜印，竖印等，总之，POINT 函数用途很大。

⑨语句 PSET：此语句能在屏幕上画出一个点。解释程序用随机函数产生 PSET(x, y)的 X, Y 坐标，然后用 PSET 语句在屏幕上绘制出“满天繁星”。PSET 语句多用于绘制不规则图形。

⑩函数 SCREEN：函数 SCREEN 用于在文本方式下得到某一字符所在位置的特征：包括前景颜色、背景颜色、是否高亮度，是否闪烁等。进入解释程序后，请用户输入一个字符，程序自己设置它的显示特征，将其结果显示在屏幕

止。

⑩函数 **STRING \$**：此函数可将 ASCII 码值代表的字符复制成一个字符串，其长度由用户自定。解释程序将所有 ASCII 码值对应的字符（不包括控制字符）逐一布满整个屏幕。**STRING \$** 函数多用于制作文本方式下的花边或分隔线等。

二、BG 的操作方法。

BG 采用全菜单式结构，每一步操作都配有详尽明确的提示。除快速动态查询窗口菜单外，所有菜单及提示均采用中文。用户只需按中文提示进行操作即可。步骤如下：

1. 进入 BG。

用 CCDOS 2.0 以上版本启动后，出现系统提示符 A>，将装有 BG 的磁盘插入 A 驱动器中，键入 BG 屏幕上出现一只眼睛快速闪动的猫，表明已进入 BG 状态，见附图 3。键入<空格>键继续。

2. 进入软件说明页。如附图 4。用户敲<H>键可以获得一切帮助信息。

3. 进入主控菜单。

首先进主菜单入口，如附图 5。按<空格>键翻页。进入到主菜单选择页，如附图 6。用户可以根据需要选择功能。注意：如果您已看过软件说明页可以在猫头那一页敲<F>键；跳过说明页直接进入主菜单入口。

4. 进入“BASIC 基本语句介绍”

首先进入入口，如图 7 按<空格>出现如图 8 所示的实际屏幕，附图 9 是其中文翻译，不出现。用户只要按↑（或↓）窗口内的语句菜单



附图 3

IBM 个人计算机

BASIC 语言系列教育软件之一

IBM PC BASICA V2.0 版

绘图辅助教学软件 BG v1.0

开发人: 北工大计算机系 张平合 朱卫平

版权人: 电子工业出版社 <H> clp

附图 4

BASIC 语言系列教育软件之一 1989.6

BG

Basic Graphics

主 菜 单

按<空格>键继续.....

附图 5

IBM PC BASIC 及绘图辅助教学)

• ● BASIC 基本语句介绍

BASIC 多种功能演示

退出

按↑, ↓键移动小球, 按空格进入

附图 6

BASIC 语言系列教育软件之一) 1989.6

BG

Basic Graphics

BASIC V2.0 基本语句介绍

按<空格>键继续.....

附图 7

IBM PC BASIC INSTRUCTIONS

IOCTL\$

KEY

KEY\$

KILL

LEFT\$

LEN

LET

LINE

LIST

LLIST

LOAD

LOC

LOCATE

LOF

LOG

LPOS

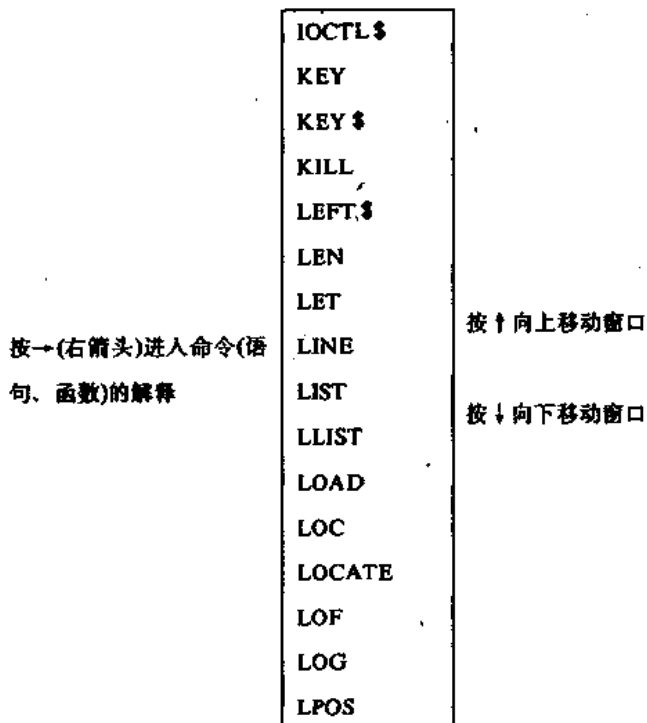
Press→ (cursor right) for a
description of this word...

Press ↑ to move up
through the list.

Press ↓ to move down
through the list.

Press <ESC> to quit

附图 8



附图 9

便快速下滚(或上滚)。窗口左边的→所指的语句项为被选语句。按下→便会在屏幕上出现该语句的解释信息。下面以 PAINT 语句为例演示一下操作方法:

(1) 用↓使快速查询窗口中的菜单项上滚, 至→指向 PAINT 语句, 如附图 10。

IBM PC BASIC INSTRUCTIONS

Press → (cursor right) for →
a description of this world
...

OCT \$
OFF
ON
OPEN
OPTION
OR
OUT
PAINT
PEEK
PEN
PLAY
PMAP
POINT
POKE
POS
PRESET

Press ↑ to move up
through the list.

Press ↓ to move down
through the list

Press <ESC> to quit

附图 10

(2) 按 → 进入附图 11, 这是基本语句说明页。因为 PAINT 是绘图语句, 它的后面带有一个解释程序, 所以按 → 将进入解释程序。如附图 12 所示, PAINT 语句的解释程序向您提出一连串的问题, 您回答之后, 进入附图 13。

语句: PAINT	说 明
格式: PAINT (X, Y) [颜色[边界色] I PAINT(X,Y)盖瓦,颜色	这是一个极其重要的绘图语句,它允许你往一个复杂的封闭图形中填颜色,及视测花纹。格式一中颜色值为 0 到 3, 缺省值为 3, 边界色是要填满颜色图形的边线颜色。格式二中盖瓦是 1 至 64 个字符的字符串,它规定了所填花纹的形状,详见解释程序。
按→进入解释程序。 按<ESC>键返回菜单	

附图 11

* PAINT 语句解释程序 *

分辨率或高分辨率(M / H)? m

背景颜色(0—15)? 4

调色板(0 or 1)? 1

盖瓦字节号(1—64)? 2

盖瓦字节 1? 12

盖瓦字节 2? 33

附图 12

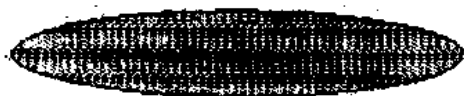
分辨率: 中(m)

背景颜色: 4

调色板: 1

盖瓦字节号: 2

盖瓦字节:



按<ESC>键退出。任按一键继续...

附图 13

(3) 任按一键 BG 返回附图 12 所示的画面，继续提问；如果在附图 13 的情况下连按两次<ESC>键，BG 返回到快速查询窗口，如附图 10 所示。此时，您可以继续查询其它语句的信息。

5. 进入“BASIC 多种功能演示”

当在如附图 6 所示的主控菜单下选择第二项时，进入附图 14。根据屏幕底行提示按一下<空格>键，红色方框则移向下一个选择项，同时发出一声短的鸣叫，当方框移到第十个选择项时，再按则返回第一项。如果你选定了某一项，则敲<ENTER>键，从而进入该项内容。下面分别介绍每一部分的内容及相应的操作方法。

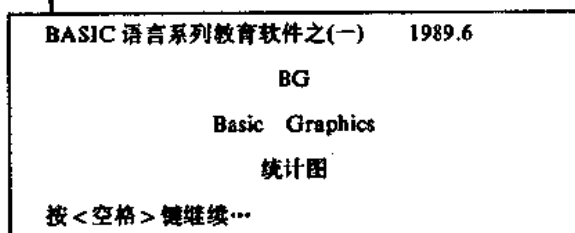
高级 basic 语言多种功能演示程序	
1 统计用图	6 计算机音乐艺术
2 三维及艺术图形	7 窗口技术
3 舞蹈家	8 动画——蜜蜂采蜜
4 CAI 辅助教学	9 太空飞船
5 公路赛车	10 退回

按<空格>移动方框，按<enter>选定

附图 14

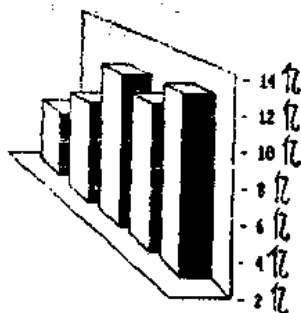
(1) 统计用图:

选择第一项, 键入<ENTER>, 进入附图 15. 按<空格>进入附图 16, 按<ESC>返回上一级菜单.



附图 15

某公司过去五年每年的销售额

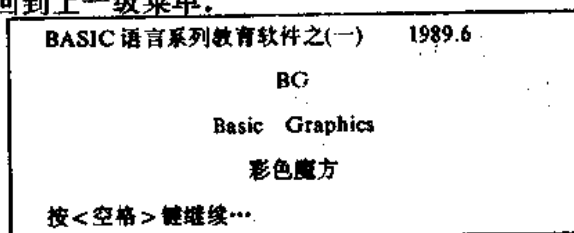


附图 16

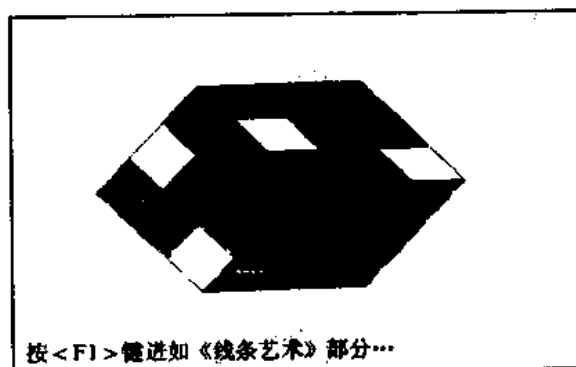
(2) 三维及艺术图形

选择第二项, 键入<ENTER>, 则显示附图 17 附图

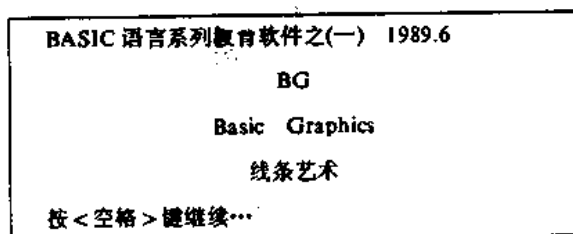
18. 按<F1>键显示附图 19~附图 22。程序全部演示完成后自动回到上一级菜单。



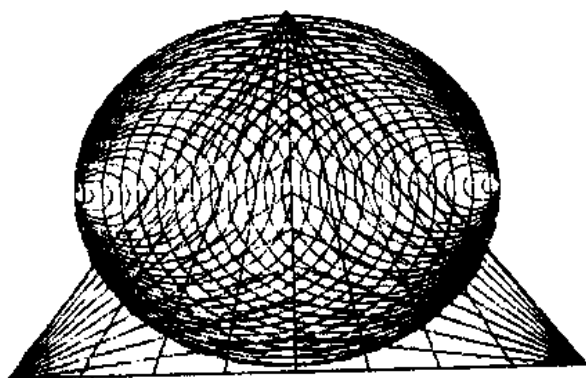
附图 17



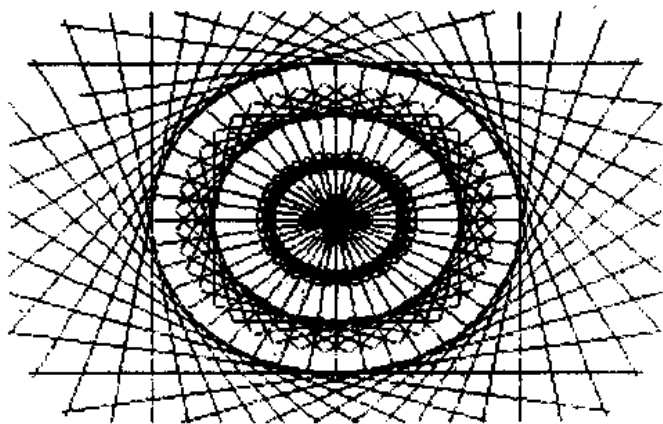
附图 18



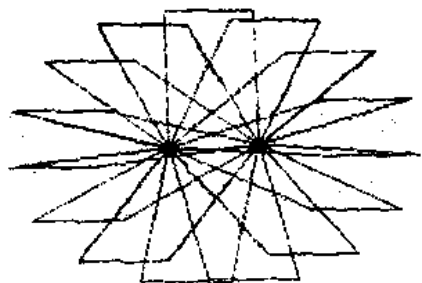
附图 19



附图 20



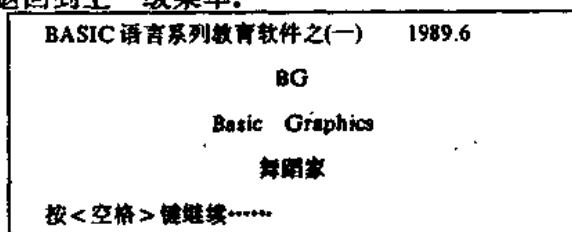
附图 21



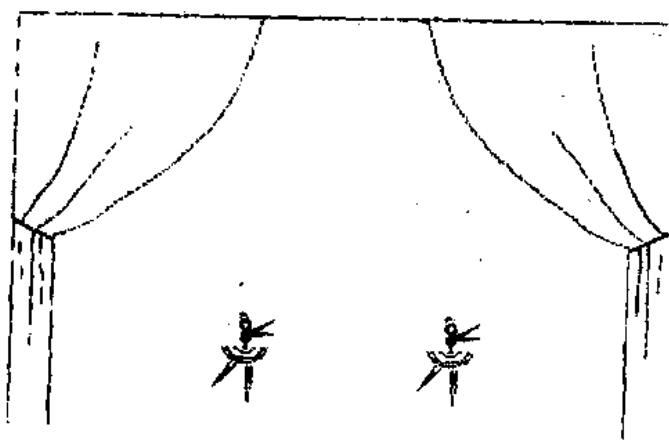
附图 22

(3) 舞蹈家。

选择第三项，键入 <enter>，出现附图 23，按 <空格> 出现附图 24。在两个小人跳动的过程中如果按 <ESC> 键，可以中断演示，返回到上一级菜单，否则小人演示完成后自动返回到上一级菜单。



附图 23



附图 24

(4) CAI 辅助教学

选择第四项，键入 <enter>。

(5) 公路赛车

选择第五项，键入 <enter>。

(6) 计算机音乐技术

选择第六项，键入 <enter>。

(7) 窗口技术

选择第七项，键入 <enter>。

窗口技术演示时间较长，如想中途退出返回上一级菜单，可键入 <ESC>。

(8) 动画片断——蜜蜂采蜜

按 <空格> 键把红色方框移到第 8 选择项，然后按

<enter> 键。

(9) 太空飞船

选择第九项，键入 <enter>，屏幕上出现一快速飞动的船。

6. 其它

(1) 热键：在 BG 中，本着介绍的目的，我们使用了多种形式的热键，如：任按一键程序继续运行；F 功能键；回车键；空格键；<ESC> 键↑、↓ 键等等。通过学习和分析 BG 中的程序，希望用户能够掌握设置热键的方法。

(2) 程序中断后的处理

如果由于用户操作不当，引行程序中断，返回到 OK 状态，您最好键入如下命令：

RUN "MAINMENU

以便返回到 BG 的环境中。



海渡数据 0023214

参 考 文 献

1 赤松義幸著，“パソコン图形处理テクニック”，诚文堂新光社，1984。

2 永山嘉昭，“3 次元グラフィックス入门”，日刊工业新闻社，1986.11。

3 桜田幸嗣等，“グラフィック・ワークブック”アスキー出版局，1984。

4 汤沢清，“パソコン活用プログラム”诚文堂新光社，1983。

5 “MS-DOS PC BASIC 使用手册”，北方电脑公司信息资料部。

6 张福炎等，“IBM PC 的原理及应用”南京大学出版社，1984.11。