



● 微计算机程序库丛书

微计算机BASIC语言 科学绘图50例

〔法〕R.多尼 著

田宝华 译 陈贻昆 校

电子工业出版社

TP 312
DN/1

微 计 算 机 BASIC 语 言

科 学 绘 图 50 例

〔法〕R. 多尼 著
田 宝 华 译
陈 贻 昆 校

電 子 工 業 出 版 社

内 容 提 要

本书介绍了利用微型计算机进行科学绘图的基本指令和方法。从基本几何图形、极坐标曲线、坐标变换、梁的受力分析到旋转的立体建筑图形与机械零件轴测投影等，共列举了50个例子并逐一给出程序。

本书通俗易懂，适合中等以上文化程度，具有BASIC语言知识的读者自学，也可作为大专院校及短训班的教材。对从事研究、设计的机械、航空、建筑、纺织和自动化等多种学科的工程技术人员都是适用的。

我们已将50例程序装盘，可直接在Apple I上运行，供读者选用。

JS371/24

GRAPHISME SCIENTIFIQUE
SUR MICRO-ORDINATEUR
de la 2^e à la 3^e dimension
50 applications résolues en BASIC

Robert DONY

MAISON

1984

微计算机BASIC语言

科学绘图50例

【法】R·多尼著

田宝华 译

陈贻昆 校

电子工业出版社

(北京市万寿路北口)

新华书店北京发行所发行 各地新华书店经售

铁道部第一工程局印刷厂印刷

开本: 787×1092 1/16 印张: 13 字数: 308千字

1985年11月第1版 1985年11月第1次印刷

印数: 34000册 定价: 2.50元

统一书号: 15290·183

译 者 的 话

本书根据法国 MASSON 出版社 1984 年出版的《GRAPHISME SCIENTIFIQUE SUR MICRO-ORDINATEUR》一书译出的。

这本书通俗易懂，深入浅出，适用范围广泛。介绍了目前国外微型计算机应用的一些方法和程序。

书中使用的是大家所熟悉的 BASIC 语言，介绍了绘图基础和应用实例及程序，便于掌握与上机练习。读者通过自学完全可以掌握本书的基本内容。

本书对平面和立体图形的绘制、计算机辅助设计、管理、统计和信息处理等多方面的会有所帮助。

对书中的重要公式和程序，我们进行了推证和校验。对个别不妥之处作了订正。

本书翻译承蒙沈达明教授的指导，李宝荣同志的多方面帮助，在此表示感谢。

由于译者水平所限，如果发现错误，敬请批评指正。

译者

1985. 3.

前 言

本书是为使用具有绘图功能的微计算机的人员编写的。程序编出后,在苹果Ⅰ(AppleⅠ)和王安2200(WANG2200)微计算机上试用过。但是,这些程序对于其它微计算机同样也是适用的。我们总是注意使编写的程序要尽可能的简单,并且给那些程序加有许多注释。对于其中的任何一种程序,我们每次都给出一个或几个绘图的例子。对于实数BASIC语言(Applesoft)的几个典型指令在附录中给出解释,以便使用其它型号微计算机的人便于实现必要的转换。人们可能指任何程序都不能核对输入数据的有效性。但之所以要这样做,是为了不使程序冗长累赘。如果认为有必要,使用者可自己补充。

我们认为,本书对教学也会有很大的帮助,主要是对各种科学学科而言。确实如此,长时间以来已经证实:一张草图或一幅图画要比长篇叙述或一张数字表格好得多。当今的文明不是被称为“图的文明”吗?例如,用矩阵对平面进行转换可以马上显示。此外,我们看到学生很难想象空间的图形。而这本书的一些程序可以给学生以极大的帮助。绘图计算机同样可以迅速地模拟一些现象。例如:傅立叶和马克劳林级数的收敛。当我们想到教师在黑板上画美丽的图形所耗费的时间时,绘图计算机的功能是不可忽视的。我们所选择的绘图方法是专用于绘图的计算机(例如Tektronix计算机)使用的方法。这种效率很高的方法是以窗口(WINDOW),视见区(VIEWPORT)和修剪(CLIPPING)的简单概念为基础的。我们把上述概念简化为易于使用的一些子程序,通过下面我们所选择的各种例子,将向读者作出说明。

最后,信息绘图在五十年代开始时,是在专门的很昂贵的机器上进行的。目前,一些微计算机只需几千法朗(1000法朗合人民币约260元左右),但能力比以前的计算机大得多。另外,微计算机都有绘图能力,大部分具有高分辨率的绘图能力。使我们预测到计算机的绘图技术只能以日益扩大的方式向前发展。这点使我们非常高兴!

目 录

第一章 计算机绘图基础

1. “移动”指令 (MOVE) 和 “画线”指令 (DRAW)	(1)
2. 窗口 (WINDOW) 指令	(1)
3. 视见区 (VIEWPORT) 指令	(2)
4. 两个例子	(3)
5. 如何模拟绘图指令	(6)
6. 从用户坐标转换到屏幕或绘图机坐标	(7)
7. 平面修剪问题 (CLIPPING)	(9)
8. 与窗口交点的计算	(11)
9. 画轴	(12)
10. 画刻度	(13)
11. 画视见区的框	(15)
12. “主要的”子程序	(15)
13. 小结	(18)
14. 绘图应用程序实例	(20)
A 1. 函数 $Y=F(X)$ 的绘图	(21)
A 2. 多角 形	(25)
A 3. 极坐标方程曲线	(27)
A 4. 极坐标方程的辐射状曲线	(30)
A 5. 美丽的角 形	(32)
A 6. N个纯追踪问题	(34)
A 7. 光学艺术绘图 (OP' ART 绘图)	(36)
A 8. 受集中力或分布力的梁	(37)
A 9. 参数方程曲线	(45)
A10. 外摆线和内摆线	(47)
A11. 矩形统计图和频率多角 形	(50)
A12. 函 数 $Y=\sin(X)$ 的马克劳林级数展开	(52)
A13. 高斯质 数 桌布	(54)
A14. 傅立叶级数	(56)
A15. 差拍现象	(59)
A16. 用最小二乘法进行线性回归	(60)
A17. 花边小桌布	(63)
A18. 用点或线画栅格	(64)
A19. 半对数坐标或对数坐标	(65)

A20. 从视见区的倒置看镜象作用	(69)
A21. 椭圆弹子球台的模拟	(70)
A22. 用蒙特·卡罗方法计算圆周率 π	(74)
A23. 指数函数	(76)
A24. 对数函数	(77)
A25. 高斯的洋葱曲线	(78)
A26. 滑动平均	(79)
A27. 方程实根的研究	(81)
A28. 年龄金字塔	(87)
A29. 正方形曲线	(90)
A30. 变狭窄的随机多角形	(91)
A31. 圆形统计图	(93)
A32. R—L—C 电路中阻抗 $Z=F(W)$ 的描绘	(93)
A33. R—L—C 电路中电流强度 $I=F(W)$ 的描绘	(95)
A34. 用抛物线的弧画弯矩图	(96)

第二章 二维空间绘图的数学基础

1. 平面变换	(103)
2. 比例变换	(103)
3. 对称	(104)
4. 切变	(105)
5. 单位正方形的一般变换	(105)
6. 旋转	(106)
7. 平移和齐次坐标	(107)
8. 复合变换	(108)
9. 小结	(110)
10. 绘图应用程序实例	(111)
A35. 2×2 矩阵的作用	(111)
A36. 3×3 矩阵的作用	(112)
A37. 平面变换的一般程序	(113)

第三章 一点解析几何

1. 引言	(117)
2. 圆的几种方程	(117)
3. 圆锥曲线: 椭圆—抛物线—双曲线	(119)
4. 绘图应用程序实例	(120)
A38. 三角形的外接圆	(120)
A39. 一条直线和一个圆的交点	(122)
A40. 两个圆相交	(124)
A41. 一条直线和一条抛物线相交	(126)
A42. 两条抛物线相交	(127)

A43. 凸形规则多角形和凹形规则多角形	(128)
A44. 椭圆平面变换	(130)

第四章 三维空间绘图的数学基础

1. 坐标系	(133)
2. 比例变换	(134)
3. 旋转	(135)
4. 平移	(137)
5. 对称	(137)
6. 绕任意轴旋转	(138)
7. 矩阵的逆矩阵	(139)
8. 坐标系的变换	(140)
9. 平面上的各种类型投影	(141)
10. 透视投影	(143)

第五章 三维空间形体在二维空间的图象

1. 解决问题的四个基本步骤	(145)
2. 屏幕上投影的表达式	(149)
3. 眼睛的位置和屏幕的位置对图象的影响	(150)
4. 空间修剪	(151)
5. 增加两个平行于正立面的正平面进行空间修剪	(154)
6. 绘图应用程序实例	(158)
A45. 使用子程序SR3绘制空间立方体的图象	(158)
A46. 使用程序SR3+PLANS绘制空间立方体的图象	(159)
A47. 自动调整多面体在屏幕上的投影	(161)
A48. 带有弧形曲线的机械零件的描绘	(166)

第六章 不可见的线和面

1. 引言	(175)
2. 峰值线法	(175)
3. 绘图应用程序实例	(178)
A49. 方程 $Z=F(X, Y)$ 描述的面	(178)
4. 表面可见度的测试方法	(183)
5. 绘图应用程序实例	(185)
A50. 对一个简单例子的全面分析	(185)
6. 小结	(190)

附录1: 实数BASIC的几个指令	(191)
附录2: 符号 Σ	(191)
附录3: 数量积和矢量积	(192)
附录4: 矩阵	(193)

附录5：王安2200系统的绘图子程序·····	(194)
附录6：子程序SR 3 和子程序SR 3 + PLANS的变量表 ·····	(194)
参考书目 ·····	(199)

第一章 计算机绘图基础

任何一张图都是由线段构成的，就象文字是由字母组成的一样。曲线是由一系列的线段首尾相接得到的。例如：一个圆是由边数很多的多边形来表现的。

线段是由两个端点决定的。为了在纸上画一线段，首先抬笔移动至线段的始端，然后把笔落在纸上移动到线段的另一端。在屏幕上画线或用绘图机画线，严格地说也是用同样的方法。第一步，电子束不亮或抬笔，把位置调到初始点。第二步，电子束亮或落笔，画线直到终点。

1. “移动”指令(MOVE)和“画线”指令(DRAW)

为了画线，就需要两种基本指令。

MOVE X, Y 和 DRAW X, Y

X和Y是有关点的直角坐标，其值取决于用户单位。

MOVE的意思为移动，DRAW的意思为画线。

例：10 MOVE 3, 3

20 DRAW 6, 5

30 DRAW 9, 1

40 DRAW 3, 3

在指令MOVE或指令DRAW后面写的变量可以有三种：

a) 常数 例：MOVE 3, 5

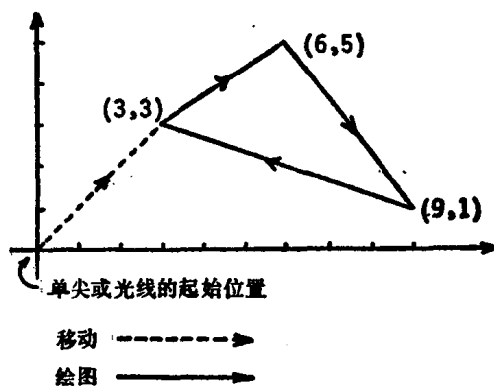
DRAW 3, 5, 8

b) 变量 例：MOVE A, B

DRAW X1, Y1

c) 表达式 例：MOVE $R * \cos(A)$,

$R * \sin(A)$

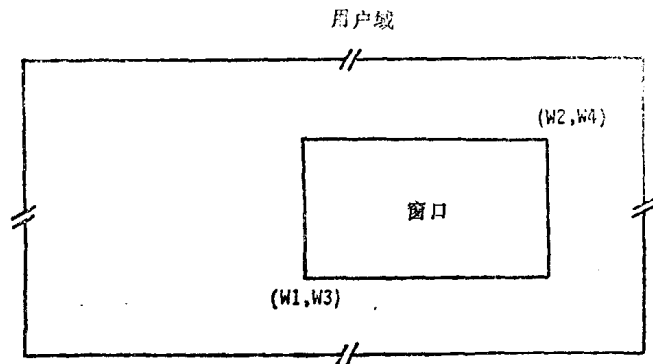


2. 窗口 (WINDOW) 指令

用户的领域是二维和三维空间。这从理论上讲是无穷的。但是，实际上将受到计算机所能接受的最高数量级的限制(例如在Apple II上为 $\pm 10^{38}$)。很显然从没有人用这样大的范围工作过。一般在用户域里取一个矩形，在这个矩形范围内进行研究。这个矩形域称为窗口。

决定窗口的四个变量W1、W2、W3、W4取决于用户单位。它们可以是年份、公里、秒、

牛顿、法朗等。



显然，在绘图之前必须让计算机知道用户所选择的工作窗口。为此，使用指令

WINDOW W 1、W 2、W 3、W 4

W 1 = 最小横坐标 W 3 = 最小纵坐标

W 2 = 最大横坐标 W 4 = 最大纵坐标

这四个变量可以是常数、变量或表达式。下面给出几个窗口的例子。

WINDOW - 5, 10, - 2.5, 14

WINDOW X 1, X 2, Y 1, Y 2

WINDOW A, B, - 5, 10

WINDOW - 180°, 360°, - 3, 4

显然，永远是 $W 1 < W 2$, $W 3 < W 4$ 。

记住：窗口所确切指出的数据范围，应该是计算机能接受的。

3. 视见区 (VIEWPORT) 指令

在纸上或在屏幕上绘图时，人们总希望不要把纸或整个屏幕占满，愿意留些地方写标题或对图作说明。人们还希望在一张纸上绘几张图。因此，需要给计算机确切指出一个矩形区，在这个矩形区中人们可以看到复制的窗口内容。这个矩形区被称为视见区 (CLOT-URE)。由下面的指令确定：

VIEWPORT V 1, V 2, V 3, V 4

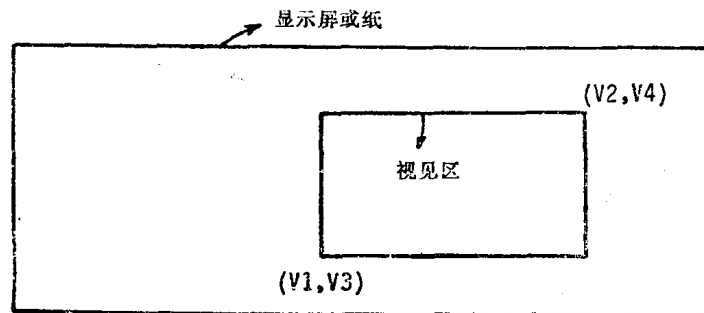
这四个变量取决于计算机的**系统绘图单位**。这意味着视见区的大小主要取决于所使用的计算机。例如在 Tektronix 4501 计算机中，最大的视见区是 100×130 个绘图单位。由于每个单位包括几个点，这个视见区可控制有 1024×780 个点的栅格。

Apple II 微计算机有一个 192×280 点的屏幕。王安绘图机可以使用 16 英寸 $\times$ 11 英寸 (40 cm $\times$ 28 cm) 的纸。因为单位增量是 $1/200$ 英寸 (0.127 mm)，由此就可以想出我们可以支配的点的数量。

视见区的四个变量可以是常数、变量或表达式。下面是几个例子：

VIEWPORT 0, 120, 10, 60

VIEWPORT A, B, C, D



记住：视见区规定能复现窗口内容的屏幕域或纸域。

4. 两个例子

1) 描绘函数 $Y = \sin(2X) \cdot e^{-0.1X}$

我们只研究 X 在区间 $[-2, 9]$ 和 Y 在区间 $[-0.6, 0.7]$ 的部分曲线。那么窗口指令是

WINDOW -2, 9, -0.6, 0.7

如果人们希望在Tektronix计算机的显示屏的右上四分之一屏幕复现图象,那么视见区的指令为:

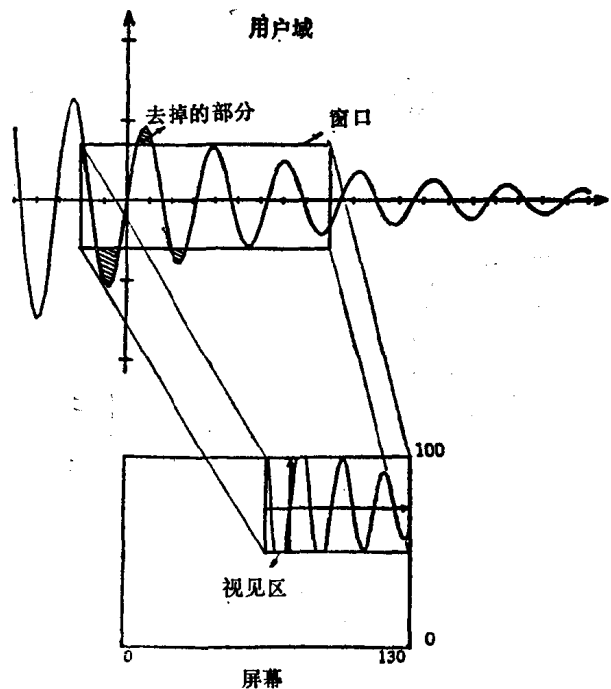
VIEWPORT 65, 130, 50, 100

下面是为Tektronix计算机用BASIC语言编写的程序。在后面可以看到用窗口和视见区概念描绘的图象。

```

100 DEF FNY(X)=SIN(2*X)*EXP(-0.1*X)
110 WINDOW -2,9,-0.6,0.7
120 VIEWPORT 65,130,50,100
130 AXIS 1,1
140 READ A,B,P
150 DATA -2,9,0.2
160 MOVE A,FNY(A)
170 FOR X=A+P TO B STEP P
180 DRAW X,FNY(X)
190 NEXT X
200 END

```



注释:

130句: 指令AXIS 1, 1 画轴并在轴上分别按1刻度,

140句: 读所研究区间和步长。

160句: 移到起始点。

170—190句: 以P为步长, 用循环在研究区间画曲线。

您看到的这个程序是简化了的。用这个程序可以画出任何曲线, 只需改变一下100句中的自定义函数和窗口及视见变量。

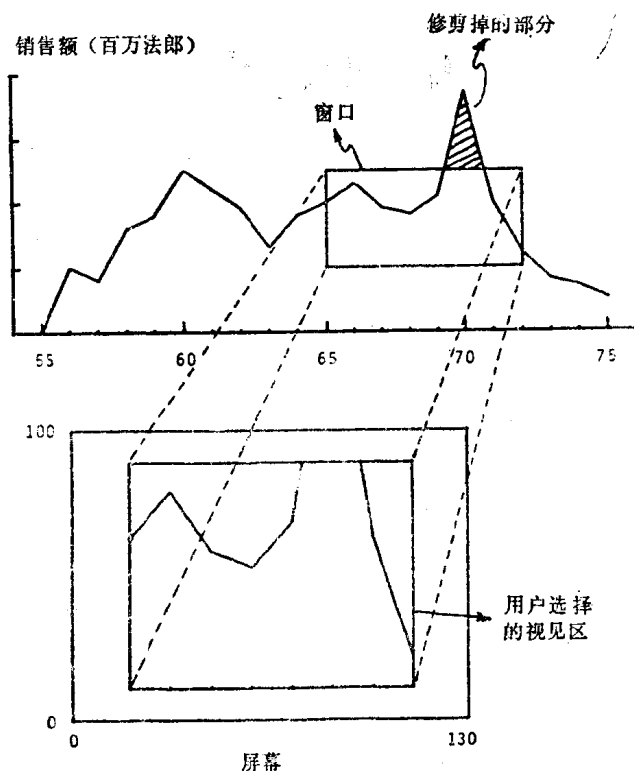
2) 设一条统计曲线表示20年中的销售额, 以百万法郎为单位。根据 Tektronix 计算机的绘图系统, 我们选择下列的窗口和视见区。

WINDOW 1965, 1972, 1, 2.5

VIEWPORT 20, 110, 10, 90

下面是 Tektronix 计算机用的BASIC 语言程序

```
100 READ W1,W2,W3,W4          190 MOVE X,Y
110 DATA 1965,1972,1,2.5      200 FOR I=2 TO N
120 READ V1,V2,V3,V4          210 READ X,Y
130 DATA 20,110,10,90        220 DRAW X,Y
140 WINDOW W1,W2,W3,W4        230 NEXT I
150 VIEWPORT V1,V2,V3,V4      240 END
160 AXIS 1,1                  250 DATA 1955,0,1956,1,1957,0.8...
170 N=21                      ( 21对数据 )
180 READ X,Y
```

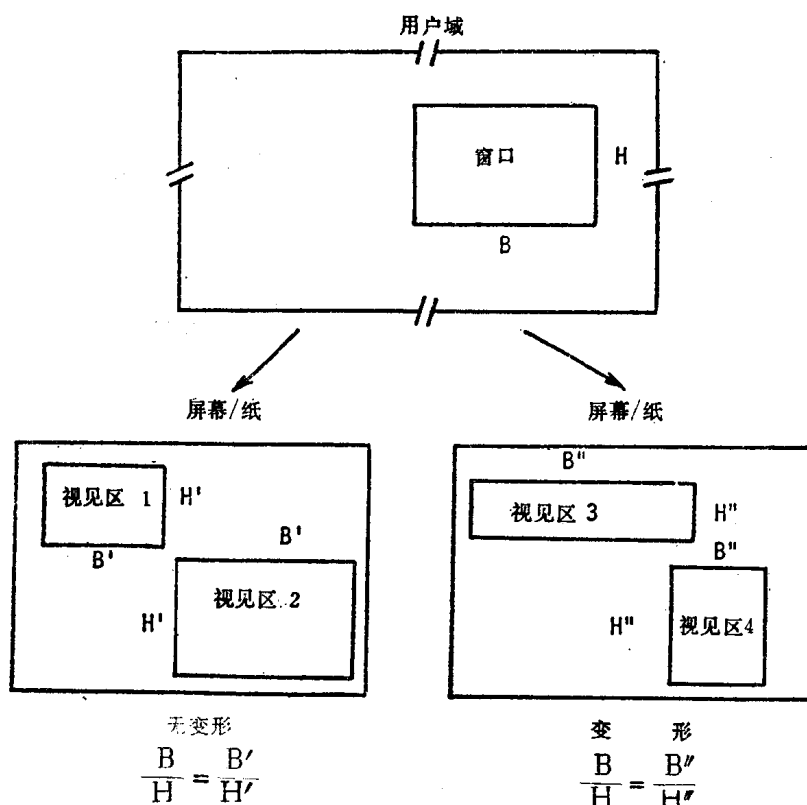


注意1:

不在窗口内的那部分图形被修剪掉了。这种修剪现象 (CLIPPING) 在信息绘图中是很重要的, 又是基本的。我们在本章的第七节中将对它进行详细讨论。

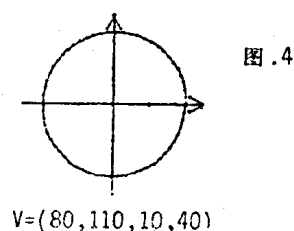
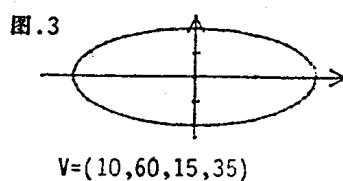
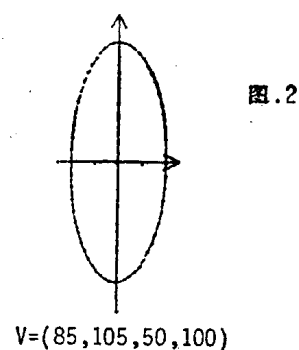
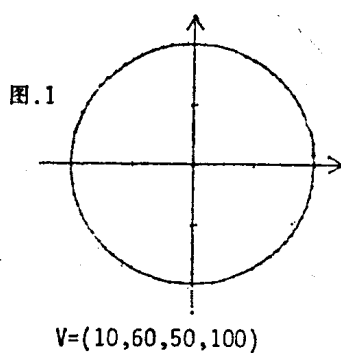
注意2:

为了得到不变形的图象, 窗口的矩形区和视见区的矩形应该是位似的。即: 窗口的底边和高之比等于视见区的底边和高之比。



例如, 设一个以一点为圆心, 以 2 为半径的圆, 显示在下面的窗口里: $W_1 = -2.5$, $W_2 = 2.5$, $W_3 = -2.5$, $W_4 = 2.5$ 。那么底 = $W_2 - W_3 = 5$, 高 = $W_4 - W_1 = 5$ 底和高之比为 1。

下图是在 Tektronix 计算机上选用不同视见区, 分别得到的结果。在 1 和 4 的情况下, 没有变形, 因为 $B/H = B'/H'$ 。相反, 在 2 和 3 的情况下, 有变形, 因为 $B/H \neq B'/H'$ 。



注意3:

选用不变的视见区和变化的窗口,可出现同样的现象。设一单位正方形,中心为原点。视见区不变总是每边为4 cm的正方形,即: $B'/H' = 1$, 对于四种不同的窗口,图中表示出不同的图象。在图1、3、和4中,没有变形,因为 $B/H = B'/H'$ 。相反,图2变了形,因为 $B/H \neq B'/H'$ 。在图4中,只是放大了(ZOOM)正方形的一部分,但是保持着比例。应指出这种放大(ZOOM)需要调用进行修剪的子程序。

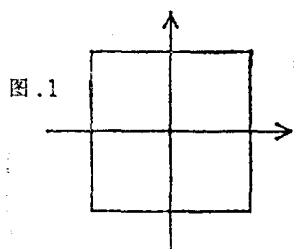


图.1

$V=(125,52.5,55,95)$

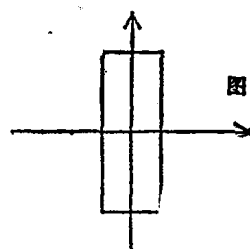


图.2

$V=(77.5,117.5,55,95)$

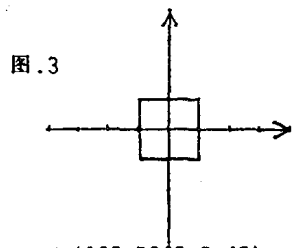


图.3

$V=(125,52.5,5,45)$

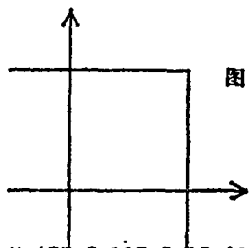


图.4

$V=(77.5,117.5,15,45)$

•上图4应为: $V=(77.5,117.5,5,45)$ ——译注

对上图进行认真观察可得出几个要点:

- 窗口扩大, 视见区不变, 图形缩小。
- 更重要的是, 窗口缩小, 视见区不变, 图形扩大。

根据后面这条性质, 对图的任何一部分可以进行放大(ZOOM)。确切地说这是一个准确的放大。因为计算重新进行, 得到了更详尽的数据。这不象在某些计算机上放大函数图象那样带着误差。在应用实例A27中, 您将看到一个很好的放大的例子。

5. 如何模拟绘图指令

用MOVE、DRAW、WINDOW和VIEWPORT这四个基本指令, 我们能绘出所有常用的图象。这些指令是Tektronix4501计算机的BASIC软件的一部分。这样从绘图的观点看, 软件就非常有力, 而且比我们见过的都要丰富。在这个软件中还可找到下面的指令:

AXIS U, V; 画X轴和Y轴, 按U和V为单位刻度。

RDRAW和RMOVE; 继续画线和继续移动。

ROTATE; 用我们选择的旋转方法进行图象显示。

遗憾的是目前便宜的微型计算机没有绘图指令。而在Apple II上则有:

HYPLOT X, Y 把电子束调到坐标点(X, Y), 使这点亮。

HYPLOT X1, Y1 TO X2, Y2 从点(X1, Y1)到(X2, Y2)画线。

王安2200计算机系统中, 有独特的指令PLOT《X, Y, U》或PLOT《X, Y, D》, 其中U的意思是UP (抬笔), D的意思是DOWN(落笔)。MOVE和DRAW同样不使用用户单位, 而使用系统绘图单位。从一个系统到另一个系统, 系统绘图单位有变化, 这就要求对每个研究的新问题进行必要的单位转换。

另外, WINDOW和VIEWPORT基本指令在小系统中不具备。因此也没有修剪问题。于是, 我们要编写出各种不同的子程序, 以弥补绘图能力不足的缺陷。我们列举了Apple II上用的那些子程序。这些子程序也易于适应其它的绘图系统。下面是子程序的内容:

S. R. 0: 初始准备: 把用户单位(FENETRE)和屏幕或绘图机的单位(CLOTURE)联系起来。

S. R. 1: 这是模拟MOVE和DRAW的主要子程序。这个子程序依赖修剪和画线子程序的帮助。

S. R. 2: 描绘COHEN和SUTHERLAND算法, 该算法在第七节中将给予详细说明。修剪以前用此子程序完成二进制编码。

S. R. 3: 计算线段与窗口边的那些可能相交点。没有这个子程序, 许多绘图是不可能的。

S. R. 4: 把要画在屏幕或纸上的点的坐标进行转换并画线。如果确实没有修剪问题, 可直接使用这个子程序代替S. R. 1子程序。在后面的例子中我们将会看到这一点。

S. R. 5: 按需要画轴。

S. R. 6: 按需要在轴上画刻度。

S. R. 7 按需要围绕视见区画框。

我们相信, 借助上述这些子程序, 可以作出所想要的所有的绘图。后面给出的例子和应用可予以证实。

6. 从用户坐标转换到屏幕或绘图机坐标

为了把选定的窗口(W1、W2、W3、W4、)在希望的视见区(V1, V2, V3, V4)上表现出来, 计算机应该认识计算点的屏幕坐标(XE, YE)的数学表达式。点的屏幕坐标(XE, YE)和用户坐标(XR, YR)是相对应的。下面就是建立的表达式。

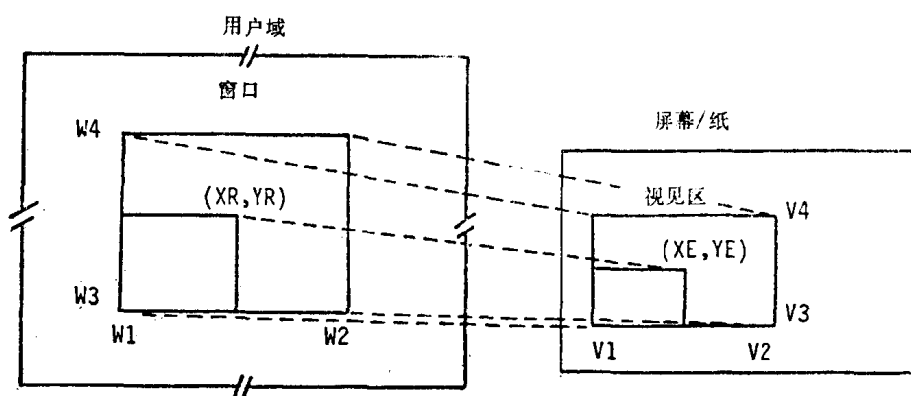
利用THALES理论, 得出:

$$\frac{XR - W1}{W2 - W1} = \frac{XE - V1}{V2 - V1} \text{ 和 } \frac{YR - W3}{W4 - W3} = \frac{YE - V3}{V4 - V3}$$

$$XE = \frac{(XR - W_1)(V_2 - V_1)}{W_2 - W_1} + V_1$$

$$YE = \frac{(YR - W_3)(V_4 - V_3)}{W_4 - W_3} + V_3$$

$$\begin{cases} XE = \frac{XR(V_2 - V_1) - W_1(V_2 - V_1) + V_1(W_2 - W_1)}{W_2 - W_1} \\ YE = \frac{YR(V_4 - V_3) - W_3(V_4 - V_3) + V_3(W_4 - W_3)}{W_4 - W_3} \end{cases}$$



$$XE = \frac{XR(V_2 - V_1) - W_1V_2 + V_1W_2}{W_2 - W_1}$$

$$YE = \frac{YR(V_4 - V_3) - W_3V_4 + V_3W_4}{W_4 - W_3}$$

$$XE = \frac{V_2 - V_1}{W_2 - W_1}XR + \frac{V_1W_2 - V_2W_1}{W_2 - W_1}$$

$$YE = \frac{V_4 - V_3}{W_4 - W_3}YR + \frac{V_3W_4 - V_4W_3}{W_4 - W_3}$$

如果记:

$$A_8 = \frac{V_2 - V_1}{W_2 - W_1}$$

$$B_8 = \frac{V_1W_2 - V_2W_1}{W_2 - W_1}$$

$$A_9 = \frac{V_4 - V_3}{W_4 - W_3}$$

$$B_9 = \frac{V_3W_4 - V_4W_3}{W_4 - W_3}$$

可得到下面的式子:

$$\begin{cases} XE = A_8 \times XR + B_8 \\ YE = A_9 \times YR + B_9 \end{cases}$$