

通用函数曲线作图程序

邓文华 梅志红

(电气工程系)

摘 要

本文用 TURBO C 2.0 编写了一个通用函数曲线的作图程序. 用户只须输入欲绘制曲线的函数表达式, 自变量 X 的变化范围和步长, 本程序就可以自动绘出相应曲线. 无论怎样修改函数表达式, 都不用改动程序, 实现了“万线一程”的设想.

关键词: 函数曲线; 字符串; 双目运算符; 翻译算法

0 引 言

利用计算机绘制各种函数曲线, 是许多用户经常遇到的问题. 一般的作法是根据给定曲线的函数关系式, 编制一个相应的作图程序. 这种做法的局限性是显而易见的, 函数关系改变, 程序就得做相应的改变. 这样的程序基本上不能交流. 为了克服这一弊病, 不少软件采用了“曲线库”的方法, 即编出若干种常用工程(函数)曲线作图子程序, 置于“曲线库”中, 通过交互方式, 供用户作图时选择. 上述两种方法都是基于“一线一程”模式, 如“曲线库”方法, 遇到库中没有的函数曲线作图要求, 仍是无能为力. 因而, 此类软件很难进行推广应用. 如果能编制一个通用函数曲线作图程序, 使用时, 用户只需输入欲绘制函数曲线的表达式及必要的坐标参数, 计算机就可自动绘制出相应的曲线, 无论怎样修改函数表达式, 都不用改动程序, 这将极大地方便用户, 也利于推广交流. 本程序即以上述想法为设计目标, 实现了“万线一程”的目标.

1 函数表达式的翻译算法

要想根据屏幕输入函数表达式绘制相应的曲线, 关键问题是找到表达式(实际是一字符串)转换成计算机的操作序列的方法, 这种转换, 就称为翻译算法, 它本是各种编译程序的主要功能, 因此, 我们可借鉴编译原理, 构造自己的函数表达式翻译算法. 本程序采用的是一遍翻译算法, 即表达式从左至右一遍扫描, 边扫描, 边分析扫描对象, 并根据对象的不同性质进行不同的操作. 扫描完毕, 即可求出相应的函数值.

本文于 1994 年元月 22 日收到

扫描算法如图 1 所示:

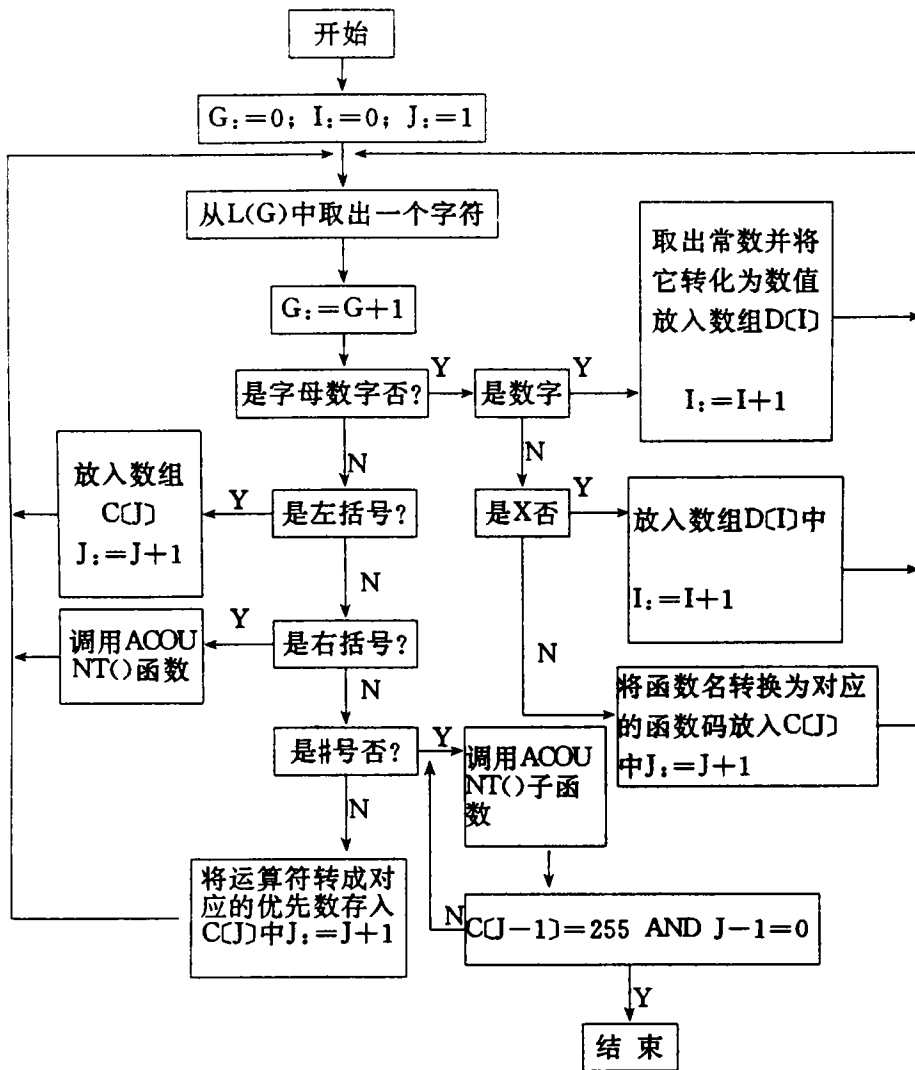


图 1

说明:

1. $L(G)$: 表达式字符串当前子串首字符;
2. #: 表达式字符串结束标志;
3. \$: 输入表达式时的结束标志;
4. 运算符包括+, -, *, /,% (取模) 5个双目运算符;
5. 本算法用到两个数组, 一个为数据数组 $D(I)$, 用于存放常数, X 变量当前值及运算的中间结果, 最后结果; 一个为符号数组 $C(J)$, 用于存放运算符, 标准函数符及分层用的左括号;

- 6. C [0] 置为 255 作为运算结束标志；
- 7. ACOUNT ()：双目运算符函数，其算法为：
 - (1) 从数据数组 D 中取出两个操作数；
 - (2) 从符号数组 C 中取出一个运算符；
 - (3) 做指定的双目运算；
 - (4) 结果放入数据数组 D [I] 中，返回。

2 程序实现时的几个问题及对策

表达式翻译这一核心算法解决后，还有一些问题需要解决，如怎样区分表达式中各种不同的成分？怎样实现标准函数的运算？怎样将求得的函数计算点 (X, Y) 转换成屏幕位置坐标 (XS, YS)？针对这些问题，本程序采取如下对策：

(1) 表达式不同成分的识别. 对初等函数表达式进行分析可知，一切初等函数不外乎都是由常数、函数、自变量、括号及运算符这几部分组成，为了简化成份识别，人为规定自变量用 X 表示，括号一律用圆括号，表达式书写遵照计算机的规定，这样，当对表达式扫描时，可做如下表处理：

表 1 表达式成分识别

L[G]	识 别	分离处理	结束条件
0..9	常数首字符	常数串利用 ATOF 把它转换成数值	非 0..9 和非.
A..Z 不含 X	函数名首字符	逐字节取函数名	非 A..Z
X	自变量	用循环变量 X 当前值代换	下一字符
(	分层符左括号	取相应的 ASCII 码	下一字符
+, -, *, /, %	运算符	取相应的 ASCII 码	下一字符
#	表达式结束符		全式结束

(2) 标准函数运算. 首先要构造一张函数名→函数码对照表，如：

函数名	函数码	函数名	函数码
fabs (绝对值)	129	sqrt	134
sin	130	atan	135
cos	131	floor (取整)	136
log	132	exp	137
Tan	133		

其次设计一个求值过程 yvalue () 它的功能是：根据给定的函数符作相应的函数运算，为了区别于双目运算符，人为规定双目运算符从 00 开始编码，即 +, -, *, /, % 对应于 0, 1, 2,

3,4.

(3) 函数计算点与屏幕显示点的坐标转换. 根据自变量 X 算出的 Y 而这个点 (X, Y) 不能直接显示到屏幕上, 需要进行必要的转换.

下面以 EGA 640 * 350 图形方式为例, 说明其转换方法.

H : 横坐标(X 轴)总刻度, 本例为 20.

V : 纵坐标(Y 轴)总刻度, 本例为 10.

转换公式:

$$X_s = \text{floor}(0x + X * bx);$$

$$Y_s = \text{floor}(0y + Y * by);$$

其中 $(0x, 0y)$ 为用户屏幕所建坐标的原点, 本例为 $(320, 199)$; bx, by , 为坐标转换系数, 本例中

$$bx = 640/H = 640/20 = 32; \quad by = -350/10 = -35,$$

对每个计算点经上述转换后即可显示于屏幕.

3 程序使用说明

本程序用 TURBO C 2.0 编成, 在 AST386 机上通过运行(显示器为 EGA 640/*350), 适用于任意初等函数. 由于篇幅所限, 本程序中暂只选择了常用的九个标准函数, 如应用时碰到此程序中没有的标准函数, 你可以方

便地在函数 YVALUE() 中的 CASE 语句中加上它的第一字母, 然后在 FUNVAL() 函数中加上具体运算函数即可. 本程序代码短, 执行速度块, 操作方便(汉字提示输入信息)灵活. 如输入表达式为 $1 + \sin(2 * X) * 4$ 时只要

按提示输入: $1 + \sin(2 * X) * 4$
\$

然后按提示输入自变量范围, 如
-10, 10

输入步长, 如 0.01

即可.

作图实例:

例 1 输入函数表达式 $1 + \sin(2 * x) * 4$ (见图 2)

请输入函数表达式?

$1 + \sin(2 * x) * 4$ # \$

请输入函数变量 x 的初值 x_0 , 终值 x_n 及变化步长 dx ?

-10

10

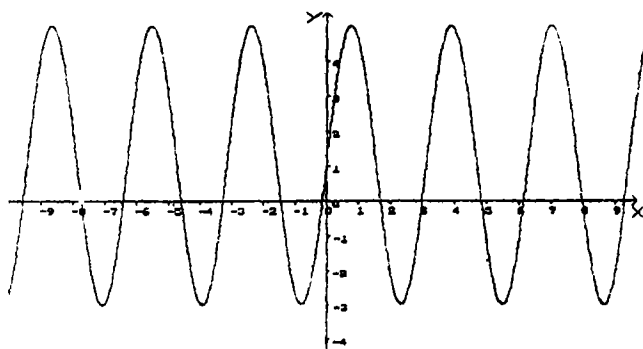


图 2

0.01

例2 输入函数表达式 $2 * x * x + 1$
1(见图3)

请输入函数表达式?

$2 * x * x + 1 \# \$$

请输入函数变量 x 的初值 x_0 , 终
值 x_n 及变化步长 dx ?

-10

10

0.01

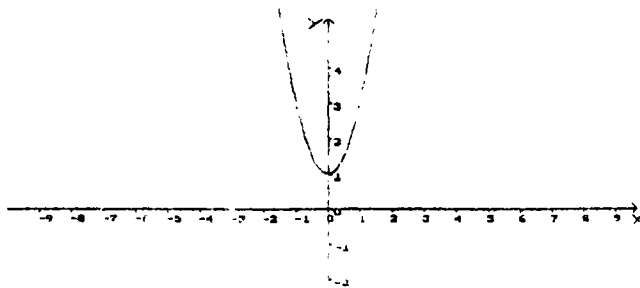


图 3

例3 输入函数表达式 $4 * x - 2$
(见图4)

请输入函数表达式?

$4 * x - 2 \# \$$

请输入函数变量 x 的初值 x_0 , 终
值 x_n 及变化步长 dx ?

-10

10

0.01

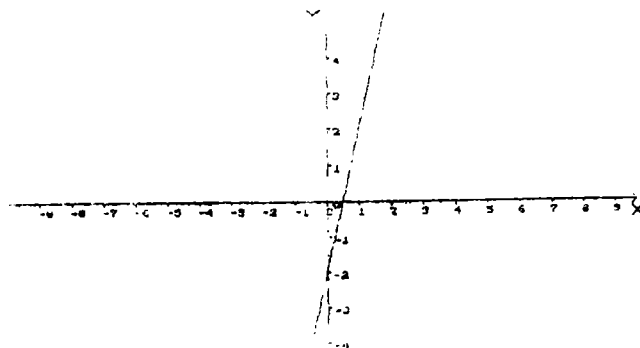


图 4

附:主程序清单

```
#include "stdlib.h"
#include "stdio.h"
#include "math.h"
#include "ctype.h"
#include "string.h"
#include "graphics.h"
float x,y,ox,oy,h,xo,xn,dx,d[100];
int c[100],g,k,p,w,j,i,n,bx,by,xs,ys;
char L[200];
main() /* 主程序,通用函数曲线作图程序 */
{
    int pl=-10,p2=-5;
    char ch,s[100];
    int grapdriver=DETECT,graphmode=0;
    i=0;h=0;g=0;c[0]=255;
    initgraph(&grapdriver,&graphmode,"c:\\tc\\"); /* 初始化图形系统 */
```

```

setbkcolor(0);          /* 设置背景颜色 */
setcolor(4);            /* 设置文本颜色 */
cleardevice();          /* 清屏幕 */
setactivepage(0);       /* 激活 0 页显示 */
setvisualpage(0);
printf("      请输入函数表达式? \n");
while ((ch=getchar())!= '$ ') {L[i]=ch;i+=1;}
/* 表达式按一个个字符存入数组: */
w=i-1;                  /* W 为表达式字符个数 */
for (i=0;i<=5;i++) d[i]=0;
printf("请输入函数变量 x 的初值 xo,终值 xn 及变化步长 dx? \n");
scanf("%f%f%f",&x0,&xh,&dx);
cleardevice();
ox=320;oy=199;          /* 定义屏幕原点坐标 */
bx=640/20;              /* 把横坐标分为 20 等分 */
by=350/10;             /* 把纵坐标分为 10 等分 */
x=x0;yvalue();
xs=floor(ox+x * bx);    /* 坐标转换 */
ys=floor(oy-y * by);
line(0,199,630,199);    /* 画横纵坐标线 */
line(320,10,320,349);
line(315,15,320,10);    /* 标 X,Y 并画出轴上箭头 */
line(320,10,325,15);
line(625,206,635,216);
line(635,206,625,216);
line(625,194,630,199);
line(630,199,625,204);
line(300,10,305,15);
line(315,10,300,20);
n=3;
for(p=32;p<640;p+=32)   /* 标坐标刻度并画出刻度线 */
{line(p,199,p,202);
 pl+=1;
 sprintf(s,"%d",p1);
 outtextxy(p-2,206,s);}
for(P=339;P>35;p-=35)
{line(320,p,320+n,p);
 P2+=1;

```

```
sprintf(s, "%d", p2);
outtextxy(326, p, s);}
moveto(xs, ys);
while(x<=xn)    /* 随给定的 X 步长递增, 算出对应的 Y 值并转换成屏幕坐标 */
{yvalue();      /* 调用求函数值子函数 */
xs=floor(ox+x * bx);
ys=floor(oy-y * by);
lineto(xs, ys); x=x+dx;
}
getch();        /* 输入任一字符退出 */
closegraph();
}
/* 主程序完 */
```

参 考 文 献

- [1] 王鹰翔. Turbo C 高级程序设计. 北京: 宇航出版社, 1992
- [2] 金成植. 编译原理. 北京: 高等教育出版社, 1987

A Program for Drawing General Function Curve

Deng Wenhua Mei Zhihong

ABSTRACT

This paper presents how to design a program for drawing general function curve. You can only input a function expression which you want to draw, the range of variable X and X's step that the program would draw the function curve automatically. No matter how you change the function expression, you need not to rewrite the program. It realizes a dream that "one program, but any function curve".

Key words: Function curve; String; operator; Translation algorithm