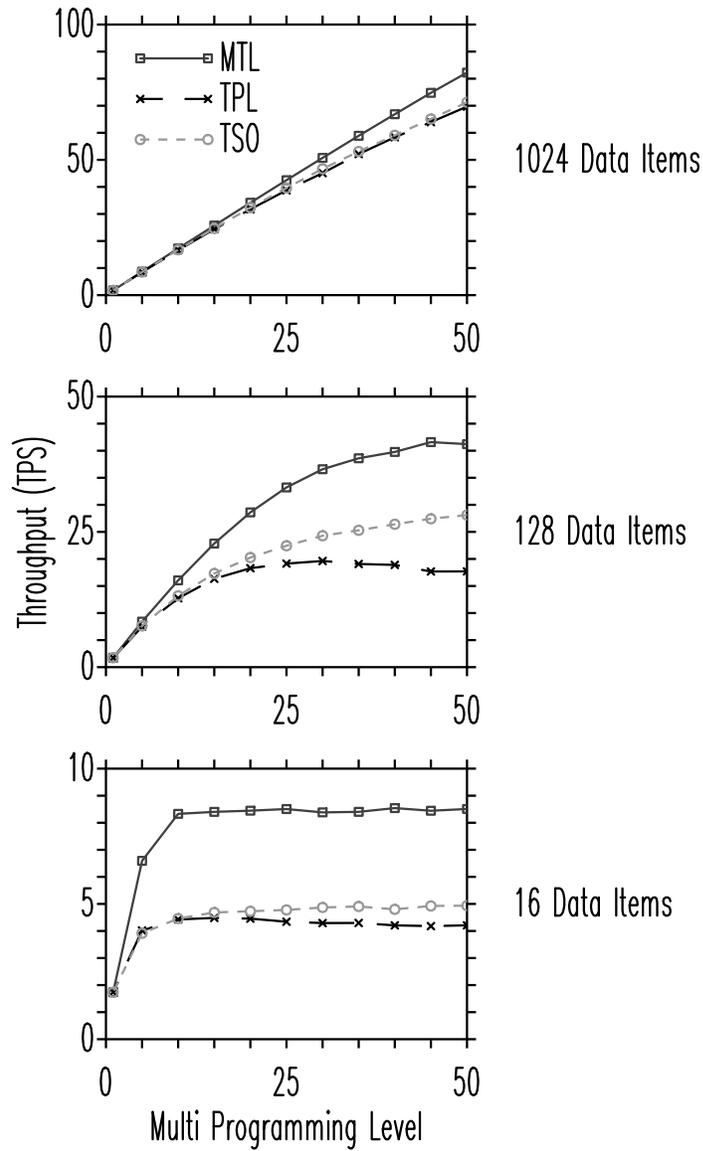


5.11 multplot.spt



```
#include <spot.h>

int i,j;
double calc[11][4];
double *data;

double calc2[11][4];
double *data2;

double calc3[11][4];
```

```

double *data3;

main()
{
    /* read in data files */
    readdata("demo\mult1.dat",data);
    readdata("demo\mult2.dat",data2);
    readdata("demo\mult3.dat",data3);

    /* average batch values */
    for (i = 0; i < 11;i++)
    {
        for (j = 0;j < 10;j++)
        {
            calc[i][0] += data[i * 10 + j][0];
            calc[i][1] += data[i * 10 + j][1];
            calc[i][2] += data[i * 10 + j][2];
            calc[i][3] += data[i * 10 + j][3];
            calc2[i][0] += data2[i * 10 + j][0];
            calc2[i][1] += data2[i * 10 + j][1];
            calc2[i][2] += data2[i * 10 + j][2];
            calc2[i][3] += data2[i * 10 + j][3];
            calc3[i][0] += data3[i * 10 + j][0];
            calc3[i][1] += data3[i * 10 + j][1];
            calc3[i][2] += data3[i * 10 + j][2];
            calc3[i][3] += data3[i * 10 + j][3];
        }
        calc[i][0] /= 10;
        calc[i][1] /= 10;
        calc[i][2] /= 10;
        calc[i][3] /= 10;
        calc2[i][0] /= 10;
        calc2[i][1] /= 10;
        calc2[i][2] /= 10;
        calc2[i][3] /= 10;
        calc3[i][0] /= 10;
        calc3[i][1] /= 10;
        calc3[i][2] /= 10;
        calc3[i][3] /= 10;
    }

    abox(8,6,5,19.25);
    ascale(YAXES,0,100);
    ascale(XAXES,calc,0);

    /* set(YMULT,10);*/
    set(FONTMULT,0.7);
    set(TICKLENGTH,-0.2);

    tickmarks();
    ticklabel();
}

```

```
text(14.03,22.25,"1024 Data Items");

set(PLOTTYPE,SYM_LINES);
set(AXESCLIP,ON);
set(CURSYPMBOL,OSQUARE);

set(LINECOLOUR,RED);
set(LINESTYLE,SOLID);
drawdata(calc,0,1);

set(CURSYPMBOL,CROSS);
set(LINECOLOUR,BLACK);
set(LINESTYLE,DASHED);
drawdata(calc,0,2);

set(CURSYPMBOL,OCIRCLE);
set(LINECOLOUR,GREEN);
set(LINESTYLE,DOTTED);
drawdata(calc,0,3);

/* the second data set */
set(LINECOLOUR,BLACK);
set(LINESTYLE,SOLID);

abox(8,6,5,11);
ascale(YAXES,calc2,1);
ascale(XAXES,calc2,0);

tickmarks();
ticklabel();

label(LEFT,"Throughput (TPS)");
text(14.03,14,"128 Data Items");

set(CURSYPMBOL,OSQUARE);

set(LINECOLOUR,RED);
set(LINESTYLE,SOLID);
drawdata(calc2,0,1);

set(CURSYPMBOL,CROSS);
set(LINECOLOUR,BLACK);
set(LINESTYLE,DASHED);
drawdata(calc2,0,2);

set(CURSYPMBOL,OCIRCLE);
set(LINECOLOUR,GREEN);
set(LINESTYLE,DOTTED);
drawdata(calc2,0,3);

/* the third data set */
set(LINECOLOUR,BLACK);
```

```
set(LINESTYLE,SOLID);

abox(8,6,5,2.75);
ascale(YAXES,0,10);
ascale(XAXES,calc3,0);

tickmarks();
ticklabel();

text(14.03,5.75,"16 Data Items");

set(CURSYMBOL,OSQUARE);

set(LINECOLOUR,RED);
set(LINESTYLE,SOLID);
moveto(5.76,24.5);
symbol(OSQUARE);
rlineto(1.5,0);
symbol(OSQUARE);
drawdata(calc3,0,1);

set(CURSYMBOL,CROSS);
set(LINECOLOUR,BLACK);
set(LINESTYLE,DASHED);
moveto(5.76,23.6);
symbol(CROSS);
rlineto(1.5,0);
symbol(CROSS);
drawdata(calc3,0,2);

set(CURSYMBOL,OCIRCLE);
set(LINECOLOUR,GREEN);
set(LINESTYLE,DOTTED);
moveto(5.76,22.7);
symbol(OCIRCLE);
rlineto(1.5,0);
symbol(OCIRCLE);
drawdata(calc3,0,3);

set(LINECOLOUR,BLACK);
set(LINESTYLE,SOLID);

label(BOTTOM,"Multi Programming Level");
text(7.53,24.5,"MTL");
text(7.53,23.6,"TPL");
text(7.53,22.7,"TSO");
}
```

Chapter 6

Splot Definitions

Contents of Splot.h Constant Definition File

```
/* splot.h */

#ifndef SPLOT_H
#define SPLOT_H

/* general defines */
#define OFF 0
#define ON 1
#define TRUE 1
#define FALSE 0

/* define line styles */
#define SOLID 0
#define DASHED 1.0,0.5
#define DOTTED 0.2 0.2
#define DOTDASH 1.0,0.5,0.25,0.5

/* page size in cm */
#define XLENGTH 20.00
#define YLENGTH 26.00

/* define graphics draw op */
#define REP 0
#define AND 0x08
#define OR 0x10
#define XOR 0x18

/* define colours */
#define INVIS -1
#define BLACK 0
#define BLUE 1
#define GREEN 2
#define CYAN 3
#define RED 4
#define MAGENTA 5
```

```
#define BROWN 6
#define WHITE 7

/* define the axes labels */
#define LOWER 0
#define BOTTOM 0
#define UPPER 2
#define TOP 2
#define LEFT 1
#define RIGHT 3
#define XAXES 4
#define YAXES 5

#define CENTER 4
#define XCENTER 1.7e308
#define YCENTER 1.7e308

/* define axes types */
#define LINEAR 0x11
#define LOGX 0x12
#define LOGY 0x21
#define LOGLOG 0x22
#define INVX 0x14
#define INVY 0x41
#define INVINV 0x44
#define INVXLOGY 0x24
#define LOGXINVY 0x24

/* define line cap styles */
#define BUTTCAP 0
#define ROUND CAP 1
#define PROJCAP 2

/* define line join styles */
#define MITERJOIN 3
#define MITREJOIN 3
#define ROUNDJOIN 4
#define BEVELJOIN 5

/* define fill rules */
#define NONZWIND 0
#define EVENODD 1

/* define devices */
#define DISPLAY 0
#define LASERJET 2
#define PAINTJET 3
#define PCXFILE 4
#define HP500 5
#define HP500C 6
#define OS2PMPRINT 7
```

```
/* define which column for error bars */
#define XVALS 0
#define YVALS 1

/* define default font info */
#define CELL_WIDTH 0.7 /* default font width */
#define CELL_ASPECT 2.0 /* default font height/width ratio */
#define CELL_SPACING 2.0

#define NFonts 7 /* number of available fonts */
#define NORMALTEXT 0 /* for use with array of font pointers */
#define ROMAN_S 1
#define ROMAN_C 2
#define GREEK_S 3
#define GREEK_C 4
#define ITALIC_C 5
#define SYMBOL_S 6
#define SIMPLEX 0
#define COMPLEX 1

/* define symbol names */
#define OCIRCLE 0
#define OSQUARE 1
#define OTRIANGLE 2
#define ODIAMOND 3
#define OSTAR 4
#define OARROW 5
#define PLUS 6
#define CROSS 7
#define MULT 8
#define CIRCLE 9
#define SQUARE 10
#define TRIANGLE 11
#define DIAMOND 12
#define STAR 13
#define ARROW 14

/* define plot types */
#define LINES 0
#define SYMBOLS 1
#define SYM_LINES 2

/* define set options */
#define AXESCLIP 0
#define AXESTYPE 1
#define CURSYMBOL 2
#define FILLRULE 3
#define FLATNESS 4
#define FONT 5
#define FONTASPECT 6
#define FONTDIR 7
#define FONTMULT 8
```

```
#define FONTSPACE 9
#define FONTWIDTH 10
#define LABELMARG 11
#define LINECAP 12
#define LINECOLOUR 13
#define LINEJOIN 14
#define LIFESTYLE 15
#define LINEWIDTH 16
#define MITERLIMIT 17
#define PAGEROT 18
#define PATTOFF 19
#define PLOTTYPE 20
#define SCALEALL 21
#define SCRIPTSCALE 22
#define SCRIPTSHIFT 23
#define SYMMULT 24
#define TICKLMARG 25
#define TICKLENGTH 26
#define XMULT 27
#define XRANGE 28
#define XSHIFT 29
#define YMULT 30
#define YRANGE 31
#define YSHIFT 32
#define LINECOLOR 13
#define MITRELIMIT 17
#define PAGE_ROT 18

/* set aliases for old commands */
#define tabdata readdata
#define hputs text

#endif

/* splot.h */
```

Chapter 7

Setable Parameters

Table of Default Values for Setable Parameters

The following values are in effect upon program startup, after execution of the "reset();" command or after clicking on the "reset" menu button. Other possible pick list choices defined in `splot.h` are given in brackets.

AXESCLIP = OFF	(ON)
AXESTYPE = LINEAR	(LOGX, LOGY, LOGLOG, INVX, INVY, INVINV, INVXLOGY, LOGXINVY)
CURSYMBOL = CIRCLE	(OCIRCLE, OSQUARE, OTRIANGLE, ODIAMOND, OSTAR, OARROW, PLUS, CROSS, MULT, CIRCLE, SQUARE, TRIANGLE, DIAMOND, STAR, ARROW)
FILLRULE = NONZWIND	(EVENODD)
FLATNESS = 1	
FONT = SIMPLEX	(COMPLEX)
FONTASPECT = 2.0	
FONTDIR = 0.0	
FONTMULT = 1.0	
FONTWIDTH = 0.7	
SYMMULT = 1.0	
LABELMARG = 0.0	
LINECAP = BUTTCAP	(ROUNDCAP, PROJCAP)
LINECOLOUR = BLACK	(INVIS, BLUE, GREEN, CYAN, RED, MAGENTA, BROWN, WHITE)

LINEJOIN = MITERJOIN (BEVELJOIN, ROUNDJOIN)

LINESTYLE = SOLID (DASHED, DOTTED, DOTDASH, or enter numerical pattern)

LINEWIDTH = 0.05

MITERLIMIT = 10.0

PAGEROT = OFF (ON)

PLOTTYPE = LINES (SYMBOLS, SYM_LINES)

SCALEALL = OFF (ON)

SCRIPTSCALE = 0.5

SCRIPTSHIFT = 0.7

TICKLMARG = 0.0

TICKLENGTH = 0.3

XMULT = 1.0

XRANGE = -1.7e308, +1.7e308

XSHIFT = 0.0

YMULT = 1.0

YRANGE = -1.7e308, +1.7e308

YSHIFT = 0.0

Chapter 8

Editor

8.1 Basics

The editor integrated into Splot is a fully featured editor that is also available as a separate product. The editor is modeled after the E PC editor. This editor is however a functional superset of E with many useful additional features. The OS/2 epm editor is another editor with the same ancestors. Users of E or epm will quickly feel at home with this editor since operation is quite similar.

Highlights of this editor include unlimited undo of previous changes and corresponding step for step redo. Column editing features to allow movement of columns of text (allows easy change of indentation). Bracket matching of all sorts of brackets and C code comment delimiters. There is limited folding which displays either; only the lines starting in the first column to quickly locate the start of functions, only the changed lines or only the lines containing the search string. Furthermore, a whole ring of files can be loaded up at once allowing painless copying of text between files. One particularly nice feature is the ability to load up a set of files in a ring and rapidly cycle through them using then ctrl N and ctrl P key combinations. Text can be copied between these files by using the block and line mark commands. There are all also some features to facilitate carrying out complex repetitive editing tasks. In particular sequences of key strokes can be saved and assigned to a function key for subsequent reuse. Furthermore, any command including function key macros can be repeated a number of times automatically by entering a command multiplier. The editor is completely key configurable. A set of keystrokes can be mapped to an editor function simply by specifying the sequence in the auxiliary key map file. The combination of key mapping and key macros allow the editor to imitate the user interface of other editors or allow the construction of a custom user interface to the taste of the user.

The editor is extremely easy to use and largely self explanatory. Text is input by straight typing anywhere on the screen. Cursor movement is controlled by the arrow keys and auxiliary keypad keys as per their standard definitions as well as accelerated movement with CTRL combinations. The cursor position control keys are as follows:

Cursor Motion Keys

left arrow	left one space
right arrow	right one space
up arrow	up one line
down arrow	down one line
end	end of line
home	beginning of line

PgDn	scroll down one screen
PgUp	scroll up one screen
Center Key	fast motion toggle
CTRL left	left one word
CTRL right	right one word
CTRL up	scroll up five lines
CTRL down	scroll down five lines
CTRL end	end of file
CTRL PgDn	end of file
CTRL home	beginning of file
CTRL PgUp	beginning of file

Editor Control Keys

CTRL A	set place mark
CTRL B	bracket match {, (, [, <, or /*
CTRL C	change case
CTRL D	delete character
DEL	delete character
CTRL E	delete to end of line
CTRL F	find next occur of search string
CTRL G	go. Execute code
CTRL H	destructive backspace
CTRL I	tab
CTRL J	join next line to current line
CTRL K	cut current line at cursor
CTRL L	return to previous set place mark
CTRL M	insert new line
CTRL N	edit next file in ring
CTRL P	edit previous file in ring
CTRL Q	quit without saving file
CTRL R	redo previous undo
CTRL S	swap current and next character
CTRL T	toggle insert/replace
CTRL O	toggle insert/replace
CTRL U	undo changes to last line
CTRL V	toggle view function mode
CTRL W	write file to disk
CTRL X	delete line
CTRL BACK	delete line
CTRL Z	zap word
CTRL DEL	zap word
ALT A	alternate binary representation
ALT E	execute last macro
ALT I	insert coordinates
ALT J	join next line to current line
ALT K	enter next char literally
ALT P	position curs at prev location
ALT Q	restore a line from undo record

ALT R	re flow paragraph
ALT S	split line at cursor
ALT T	teach new macro
ALT V	view only changed lines
ALT W	view lines containing search string
ALT X	enter control character
ALT Z	zero command multiplier
ALT 0-9	enter command multiplier

The ALT P command is useful if the cursor was accidentally moved away from the region of interest with a CTRL Home or search for example. In these situations ALT P will restore the cursor to its previous location. This action is similar to using place marks except that unlike place marks these are automatically set by any command that causes the cursor to jump by more than one line. Only one previous location is remembered though.

8.2 Block manipulations

Block mark and line mark are somewhat different. Use line mark for moving, copying or deleting one or more whole lines and block mark for moving, copying, deleting, filling or overlaying columns of text. Note also that a block marked region can also be moved to the left with ctrl ^ and to the right with ctrl _ but the behaviour is different than if the ALT arrow keys are used. ALT left and ALT right (or ctrl _ and ctrl ^) are meaningless for line marked regions.

Block Manipulation Keys

ALT B	mark column block start, end
ALT C	copy marked block
ALT D	delete marked block
ALT F	fill B marked block with char
ALT L	mark LINE(s) start, end
ALT M	move marked block
ALT O	overlay B marked block
ALT U	unmark block
ALT Y	yank back deleted block
CTRL_	shift B marked block right
CTRL^	shift B marked block left

8.3 Macros

There is also a macro capability for repetitive, complex editing tasks. The editor is taught a macro by hitting the ALT T combination. After this all subsequent keystrokes are stored as the macro definition until ALT T is hit again. There is a 128 keystroke limit on the length of a macro. A macro can consist of any keystrokes except ALT T and ALT E. At the end of the macro definition the macro can be bound to a function key by hitting the desired function key. Available function keys are F1 - F12 as well as shift F1 - F12. A bound macro is subsequently executed by hitting that function key. The predefined macro for that key will of course be lost. The last learned macro whether bound or unbound can be executed

by hitting ALT E. If the macro involves a search and another item is not found the rest of the macro is not executed. If macros have been bound to function keys they will be written to the configuration file `splot.cfg` which will be created in the current directory so that the macros will be preserved in between editing sessions.

A repeat factor for a keystroke can also be entered by prefixing the command with a multiplier which is input using the ALT number keys. The multiplier can be reset to zero during input using ALT Z. This feature also facilitates repetitive editing as an operation can be repeated several times. This is particularly powerful in conjunction with macros.

8.4 Command Line

Lastly, there are a small number of commands which require typed input on the command line. The command line is reached by hitting ESC. This places the cursor at the beginning of the command line which is on the last row and ordinarily displays some status information. After ESC however text can be input here for the following functions:

Command Line Operations (requires text input so terminate line with a return)

esc goto or leave command line

Change current line
 number goto line number n
 +number down n lines
 -number up n lines

Change current column offset
 @number start at column n
 @+number scroll left n columns
 @-number scroll right n columns

Search and Replace.

find string str (/ actually any punct)

/str[/-bclr] return

replace str1 with str2

c/str1/str2[/-bcl] return

The optional search suffixes are

/- search up (down is default)
 /b search marked block only
 /c ignore case
 /l loop through all files
 /r interpret as regular expression

For regular exp the following symbols have special meaning

^ start of line

\$ end of line

. any character

quote next character

* match zero or more times
 + match one or more times
 [aei0-9] match a,e,i and 0 through 9
 [^aei0-9] match anything but a,e,i and 0 through 9
 () sub expression
 a(ab)*b matches ab aabb aababb aabababb etc.

Add other files to ring
 e name1,name2,...[-r]
 The optional edit suffixes are
 -r read only

display directory and choose one to edit
 e [*.*]
 activate this help file
 ?
 insert named file at current location
 m name
 rename current file to name
 r name
 show and modify editor configuration
 configure
 exit without saving changes
 quit
 previous command line entered
 up arrow
 next command line entered
 down arrow
 send command to operating system
 os command
 also any unrecognized string

8.5 configuration file

The startup file `spot.cfg` preserves the values selected from the configuration menu item between sessions. You can have multiple `spot.cfg` files in different directories since the current directory is searched before using the default `spot.cfg` stored in the same directory as the executable. The `spot.cfg` file is plain ASCII and can thus be edited with a text editor as an alternative means of changing the settings or the macro definitions. In particular, to change the text string inserted in the text by choosing a draw menu item the `spot.cfg` defined function macros should be edited.

The macros are defined in plain text and thus in order to accommodate control and cursor keys as well each macro keystroke is defined by a pair of letters using the following scheme. Starting in the first column of a definition must appear the key word or function key identifying the macro. This is followed by the macro body. Function keys may not appear in macro bodies. Keystroke and are encoded as follows with implied sequences in ASCII order.

Ordinary characters	prepend with _	
Control Keys	prepend with ^	^M Return, ^[esc
Alt Keys	prepend with %	
Cursor keys	prepend with #	#. Del, #0 Ins - #9 PgUp
Ctrl Cursor Keys	prepend with #	#N Del, #P Ins - #Y PgUp
Alt Cursor Keys	prepend with #	#n Del, #p Ins - #y PgUp
Function Keys	prepend with \$	\$1 - \$<
Shift Function Keys	prepend with \$	\$= - \$H

Chapter 9

Acknowledgements

I wish to express my thanks to the people that assisted me during the development of this program. First of all Alan Steele, who wrote the first version of the Hershey font drawing routine and with whom the philosophy of the program was extensively discussed. I would like to acknowledge Clare Morrison for helping with the preparation of this document and for extensive testing. I thank the people who put up with many alpha and beta versions that were far from perfect. In particular, I wish to thank Alan Steele, Scott Wilson, Clare Morrison, Yves Lacroix, Lori Lenchyshyn, Valentin Karasyuk, Simon Watkins and Darlene Brake for helpful feedback.

Index

- abox 35
- alineto 35
- arc 35
- arrowto 36
- ascale 36
- attributes, changing 11
- axes box 10, 36
 - invisible example 56
 - log example 26
- Bezier curves 37
- box 36
- C 5,7,8,35
 - library functions 11
 - abs 45
 - acos 45
 - asin 45
 - atan2 45
 - atan 45
 - atof 45
 - atoi 46
 - ceil 46
 - cos 46
 - exit 46
 - exp 46
 - fabs 46
 - floor 46
 - fmod 46
 - free 46
 - log10 46
 - log 46
 - malloc 46
 - pow 47
 - printf 47
 - print 47
 - puts 47
 - sin 47
 - sizeof 47
 - sprintf 47
 - sqrt 47
 - strcat 47
 - strcpy 48
 - strlen 48
 - tan 48
 - C, keywords 11
 - clear 36
 - clip 10, 31, 37
 - example 31
 - closepath 37
 - cmatch 31, 37
 - constants, definitions 77
 - coordinates 10, 12
 - changing 12
 - inserting 10
 - matching 31
 - relative 11
 - cursor positioning 83
 - curveto 37
 - data 37, 38, 39
 - always load 39
 - file 10
 - plotting 10
 - shifting example 53
 - drawdata 10, 24, 37, 39
 - clipping 31
 - drawing functions 35
 - drawing, example 66
 - editor 7, 9, 83
 - block mark 85
 - command line 86
 - control keys 84
 - macros 11, 85
 - error bars 10, 37
 - example 52
 - example files 49
 - exiting program 9
 - fill 9, 24, 37
 - finding elements 10
 - fit line 38
 - example 52
 - get attributes 38
 - greek 13

- gsave/grestore 24, 38
- hard copy 11
- highlighting elements 10
- implicit stroking 10, 39
- input focus 10
- inset 26
- installation 33
- interactive drawing 9, 10, 12
- interpreter 5
- italic 13
- label 13, 38
- landscape 31
- lineto 9, 39
- main 11
- making a drawing 9
- math 5, 11
 - example 63, 73
- menus 10, 11, 12
- moveto 9, 39
- newpath 10, 26, 39
- online help 10
- path 9, 37, 38, 40
- plotdata 7, 10, 39
 - clipping 31
- portrait 31
- postscript 5, 9, 11
 - encapsulated 11
- puts 38
- rarrowto 39
- readdata 7, 10, 39
- reset 40
- rlineto 40
- rmoveto 40
- rotate 31, 40
- saving file 9
- scale 26, 36, 40
- set attributes, default values 81
- set attributes 40
- set 9, 11, 38
 - functions 42
 - AXESCLIP 31, 42
 - AXESTYPE 42
 - CURSYMBOL 24, 42
 - FILLRULE 42
 - FLATNESS 42
 - FONTASPECT 13, 43
 - FONTDIR 13, 43
 - FONTMULT 13, 43
 - FONTWIDTH 13, 42
 - FONTSPACE 13, 43
 - FONT 13, 42
 - LABELMARG 38, 43
 - LINECAP 43
 - LINECOLOUR 13, 43
 - LINEJOIN 43
 - LINESTYLE 43, 43
 - LINEWIDTH 13, 43
 - MITERLIMIT 44
 - PAGEROT 31, 44
 - PATTOFF 44
 - PLOTTYPE 24, 44
 - SCALEALL 44
 - SCRIPTSCALE 13, 44
 - SCRIPTSHIFT 13, 44
 - SYMMULT 24, 38, 44
 - TICKLENGTH 41, 44
 - TICKLMARG 41, 44
 - XMULT 45
 - XRANGE 38, 42, 45.1
 - XSHIFT 45
- showpage 10, 40
- starting plot 33
- stroke 9, 24, 26, 40
- subroutine 5
- subscript 13
- superscript 13
- symbol 37
- symbols 8, 10, 24
- system requirements 33
- text, character table 17
- text 13, 26, 39, 41
- tick labels 30, 39
- tick marks 30, 39
 - length 30
 - width 30
 - example 30
- tickmarks 10, 41
- ticklabel 10, 13, 41
- translate 26, 31, 42
- units 10, 31
- using mouse 9, 10
- whereis 42
- white out 10
 - example 25
- zoom 10, 11