October 8, 1990

PLOT_PS

A PostScript graphic library for C

User manual for V 3.1

Jean-François VIBERT & Sylvain HANNETON

Département de Physiologie
Laboratoire d'Informatique Orientée Neurosciences
CHU Saint-Antoine
Université Pierre et Marie Curie
CNRS UA 1162
PARIS
FRANCE

Table of Content

Function	Action	Section
PLOT_PS	Generalities on PLOT_PS	1
PS_INIPLOT	Initialize the PostScript plotter	2
PS_ENDPLOT	Close and spools the PostScript file	3
PS_NEWPAGE	Go to the next page	4
PS_ORIGIN	Define new origine and move the pen	5
PS_ECHEL	Define scale of drawings (subject versus PostScipt units)	6
PS_SCALE	Scale of drawings	7
PS_PEN	Change the drawing pen	8
PS_ROTATE	Allows a rotation in degres	9
PS_FONT	Allows to change the font	10
PS_SIZE	Define size (in cm) of characters	11
PS_CHARA	Write a string in PostScipt space, stays at the end	12
PS_CHARS	Write a string in user space, stays at the end	13
PS_GRAY	Define a new default gray level	14
PS_LINE_WIDTH	Defines a new line width for the lines	15
PS_ARC	Draws an arc clockwise	16
PS_CIRCLE	Draws an empty circle	17
PS_DISK	Draws a filled disk	18
PS_ELLIPSE	Draws an empty ellipse	19
PS_ELLIP_DISK	Draws a filled ellipse	20
PS_MOVEMENTS	Functions to move pen and to draw	21
ps_3Dsetup	Set up a 3D reference	22
ps_3Dcalcul	Computes projection and rotation from 3D to 2D space	23
ps_3Daxes	Draws a 3D reference	24
ps_3Dbox	Draws a box in 3D (only 3 faces are represented)	25
ps_3Dbar	Draws a bar in 3D	26
ps_3Dframe	Draws a bar skeleton in 3D	27
ps_3Dclosebox	Close a box with skeleton in 3D	28
PS_POLYLINE	Draws a polyline	29
PS_CLOSED_CURVE	Draws a possibly filled surface	30
PS_BEZIER	Draws a Bezier curve	31
PS_POINT	Draws a point centered on the current position	32
PS_FRAME	Draws a frame at the current position	33
PS_WRITE	Writes a character string or a number	34
PS_AXES	Draws a two axes centered on the extremal values	35
PS_INCLUDE	Includes a PostScript file	36
PS_CODE	Includes PostScript code	37
PS_TICKED_AXES	Draws a ticked reference	38
PS_HISTO	Draws an histogramm	39

1 - PLOT_PS

NAME: PLOT_PS Generalities on PLOT_PS

DESCRIPTION:

PLOT_PS is a set of C functions designed to build PostScript files from C programs.

They all begin with the prefix `ps_` and need to have the `plot_ps.h` header file included.

The very first function to be called is `ps_iniplot`. It opens the postscript file.

PLOT_PS allows to work in both absolute (in cms) and user or subject space. To be able to work in the user space, the function `ps_echel` must be called to define this user space.

Inclusion of PS file or of plain PostScript code is also possible with PLOT_PS. The file inclusion possibility allows the inclusion of a drawing obtained with any wysiwyg interactive drawing program.

The very last function call must be `ps_endplot`. It closes the postscript file and spool it or not.

Before to use any 3D function, the `ps_3Dsetup` must be called in order to setup all the necessary parameters. These functions requires to be linked with the mathematics library.

The program using PLOT_PS must be linked with the plot_PS library and THEN the math library if the 3D functions are used.

Under unix for exemple:

```
cc -O -o my_prog my_prog.c -lplot_ps -lm
```

AUTHOR:

Jean-Francois VIBERT et Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

2 - PS_INIPLOT

NAME: PS_INIPLOT Initialize the PostScript plotter

SYNOPSIS:

```
#include <plot_ps.h>

ps_iniplot(plot_file,paper);
char *plot_file; /* File for the plot */
float paper; /* Paper width in cm */
```

DESCRIPTION:

PS_INIPLOT opens the PostScript file and draws a frame corresponding to the flag page of a line-printer. The date and time of the file production, the file name and the user name are indicated as well as a page number. This banner is output on each page by pl_newpage(). It is drawn at the bottom of the page and is 1 cm wide.

AUTHOR:

Dr Jean-Francois Vibert

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

plopen() and plclose()

3 - PS_ENDPLOT

NAME: PS_ENDPLOT Close and spools the PostScript file

SYNOPSIS:

```
#include <plot_ps.h>

ps_endplot(file,action);
char    *file;  /* nom du fichier dessin */
        action;/* Action to take about the plot file
                        s: spool and keep
                        d: spool and delete
                        f: freeze (no spool) */
```

DESCRIPTION:

PS_ENDPLOT closes the PostScript file and process it according to the action indicated by the char argument:

s: spool and keep

d: spool and delete

f: freeze (no spool)

AUTHOR:

Dr Jean-Francois Vibert

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

plopen() and plclose()

4 - PS_NEWPAGE

NAME: PS_NEWPAGE Go to the next page

SYNOPSIS:

```
#include <plot_ps.h>

ps_newpage()
```

DESCRIPTION:

PS_NEWPAGE go to the next page and redraws the banner with the page number.

AUTHOR:

Dr Jean-Francois Vibert

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

5 - PS_ORIGIN

NAME: PS_ORIGIN Define new origine and move the pen

SYNOPSIS:

```
#include <plot_ps.h>

ps_origin(xf,yf,xa,ya);
float   xf, yf, /* Absolute coordinates in cm */
        xa, ya; /* Absolute (en cm) coordinates
                   of new position */
```

DESCRIPTION:

PS_ORIGIN defines a new absolute origine. The pen if first moved to xf and yf coordinates. xa and ya corespond to the new coordinates of the pen.

All coordinates are given in centimeters (also refered as PostScipt space).

AUTHOR:

Jean-Francois VIBERT a partir de PLOT0.C de Jean-Noel Albert

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

6 - PS_ECHEL

NAME: PS_ECHEL Define scale of drawings (subject versus PostScipt units)

SYNOPSIS:

```
#include <plot_ps.h>

ps_echel(ex,ey,xs,ys,ca,sa);
float   ex, ey, /* Number of cms by user unit   */
        xs, ys, /* Absolute (en cm) coordinates
                              of user origine */
        ca, sa; /* Cos and Sin of the drawing
                             (rotation)       */
```

DESCRIPTION:

PS_ECHEL Define the scale of the drawing, and its origine in the PostScript coordinates (relative to the last call to ps_origin). Its allows also rotation of the entire drawing.

The scales must be given in centimeters by user (or subject) unit. They must have a value greater than 0.008 since the PostScript draws one centimeter using 118 steps. If not, the function returns FALSE.

The coordinates of the new origine must be given in PostScript units, i.e. in centimeters.

Cosinus and Sinus of the angle of the drawing relative to the X axis gives a rotation possibility.

Horizontal head up: cosa=1. sina=0.
Horizontal head down: cosa=-1. sina=0.
Verticalement ascending: cosa=0. sina=1.
Verticalement descending: cosa=0. sina=-1.

AUTHOR:

Jean-Francois VIBERT a partir de PLOT0.C de Jean-Noel Albert

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

7 - PS_SCALE

NAME: PS_SCALE Scale of drawings

SYNOPSIS:

```
#include <plot_ps.h>

ps_scale(x_scl,y_scl)
float   x_scl,
        y_scl;
```

DESCRIPTION:

PS_SCALE set a new global scale of the drawing, with uniform (if x_scl == y_scl) or non-uniform (if x_scl != y_scl) scaling. It acts on both the drawing and texts. it allows to change the proportion of the drawing as well as its size. To come back to the original scale, ps_scale must be called with 1./x_scl and 1./y_scl.

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

8 - PS_PEN

NAME: PS_PEN Change the drawing pen

SYNOPSIS:

```
#include <plot_ps.h>

ps_pen(k);
int k;    /* pen number (0, 1 or 2) */
```

DESCRIPTION:

PS_PEN selects the drawing pen. If the pen number (k) is not one of 0, 1 or 2, the function returns FALSE, else it returns TRUE. If ps_pen() is not called, the default pen is 0 (BLACK).

0 gives a black
1 gives a gray at 0.3
2 a light gray at 0.7.

AUTHOR:

Jean-Francois VIBERT a partir de PLOT0.C de Jean-Noel Albert

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

9 - PS_ROTATE

NAME: PS_ROTATE Allows a rotation in degres

SYNOPSIS:

```
#include <plot_ps.h>

ps_rotate(rot);
int      rot;  /* Rotation in degres */
```

DESCRIPTION:

PS_ROTATE() Allows a rotation in degres of the graph.

AUTHOR:

Jean-Francois

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

10 - PS_FONT

NAME: PS_FONT Allows to change the font

SYNOPSIS:

```
#include <plot_ps.h>

ps_font(font,hy);
char    *font;  /* font name */
float   hy;     /* Rotation in degres */
```

DESCRIPTION:

PS_FONT allows to choose a new font as the current font. The first argument is the PostScript build in font name. The size of the font is given in centimer by hy. It can be changed alone by ps_size().

Legal font names are:

AvanteGarde-Book
AvanteGarde-BookOblique
AvanteGarde-Demi
AvanteGarde-DemiOblique
Courier
Courier-Bold
Courier-Oblique
Courier-BoldOblique
Helvetica
Helvetica-Bold
Helvetica-Oblique
Helvetica-BoldOblique
LubalinGraph-Book
LubalinGraph-BookOblique
LubalinGraph-Demi
LubalinGraph-DemiOblique
Times-Roman
Times-Bold
Times-Italic
Times-BoldItalic
NewCenturySchlbk-Roman
NewCenturySchlbk-Bold
NewCenturySchlbk-Italic
NewCenturySchlbk-BoldItalic
Souvenir-Demi
Souvenir-DemiItalic
Souvenir-Light

Souvenir-LightItalic
Symbol

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

11 - PS_SIZE

NAME: PS_SIZE Define size (in cm) of characters

SYNOPSIS:

```
#include <plot_ps.h>

ps_size(hx);
float   hx;    /* Size of height
                  of characters given
                  in centimeters */
```

DESCRIPTION:

PS_SIZE defines the size of characters, given in centimeters, and acts on he current font choosen by ps_police().

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

12 - PS_CHARA

NAME: PS_CHARA Write a string in PostScipt space, stays at the end

SYNOPSIS:

```
#include <plot_ps.h>

ps_chara(x,y,j,str,hx,hy,rot);
float   x, y;    x and y coordinates
                     given in PostScipt units (cm)
int     j;       the Benson j parameter
char    *str;    the string to write
float   hx, hy;  character width (hx) and height (hy)
                     (in cm)
int     rot;     rotation in degres
```

DESCRIPTION:

PS_CHARA writes a string at the position indicated by x,y and j, with the rotation given by cosa and sina, and stays at the end of the string.

PS_CHARA replicates the Benson pcara fortran routine.

X and Y are the coordinates of the target given in PostScipt units (in centimeters). These coordinates are considered as absolute (j=0 or j=1) or relative (j=2 or j=3) according to the value of j. During the move, the pen is up (j=0 or j=2) or down (j=1 or j=3) according to the value of j.

The following constants are defined in PLOT_PS.H:

Code des deplacements de la plume

#define PL_AH 0 Absolu Plume Haute
#define PL_AB 1 Abslou Plume Basse
#define PL_RH 2 Relatif Plume Haute
#define PL_RB 3 Relatif Plume Basse

The size of characters is given in centimeters by hx (width) and hy (height) of the characters.

AUTHOR:

Jean-Francois VIBERT a partir de PLOT0.C de Jean-Noel Albert

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

All text writing functions

13 - PS_CHARS

NAME: PS_CHARS Write a string in user space, stays at the end

SYNOPSIS:

```
#include <plot_ps.h>

ps_chars(x,y,j,str,hx,hy,rot);
float   x, y;    x and y coordinates
                     given in PostScipt units (cm)
int     j;       the Benson j parameter
char    *str;    the string to write
float   hx, hy;  character width (hx) and height (hy)
                     (in cm)
int     rot;     rotation in degres
```

DESCRIPTION:

PS_CHARS writes a string at the position indicated by x,y and j, with the rotation given in degre, and stays at the end of the string.

PS_CHARS almost replicates the Benson pcars fortran routine.

X and Y are the coordinates of the target given in user units (NOT in centimeters). These coordinates are considered as absolute (j=0 or j=1) or relative (j=2 or j=3) according to the value of j. During the move, the pen is up (j=0 or j=2) or down (j=1 or j=3) according to the value of j.

The following constants are defined in PLOT.H:

Code des deplacements de la plume
#define PL_AH 0 Absolu Plume Haute
#define PL_AB 1 Abslou Plume Basse
#define PL_RH 2 Relatif Plume Haute
#define PL_RB 3 Relatif Plume Basse

The size of characters is given in centimeters by hy (height) of the characters.

AUTHOR:

Jean-Francois VIBERT a partir de PLOT0.C de Jean-Noel Albert

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

All text writing functions PS_WRITE*

14 - PS_GRAY

NAME: PS_GRAY Define a new default gray level

SYNOPSIS:

```
#include <plot_ps.h>

ps_gray(gray);
float     gray; /* gray for the subsequent drawings */
```

DESCRIPTION:

PS_GRAY() defines a new gray value for the subsequent drawings.

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

15 - PS_LINE_WIDTH

NAME: PS_LINE_WIDTH Defines a new line width for the lines

SYNOPSIS:

```
#include <plot_ps.h>

ps_line_width(width);
int     width; /* PostScript width in picas */
```

DESCRIPTION:

PS_LINE_WIDTH defines a new line width for the lines subsequently drawn. The width is given in PostScript units (picas).

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

16 - PS_ARC

NAME: PS_ARC Draws an arc clockwise

SYNOPSIS:

```
#include <plot_ps.h>

ps_arc(r,theta_1,theta_2)
float   r;      /* radius always in cms */
int     theta_1,/* begining angle */
        theta_2;/* ending angle */
```

DESCRIPTION:

PS_ARC Draws an arc clockwise begining at the current position, and coms back to the same position. The radius must be given in cms. Begining and ending arcs are given in degres clockwise from the horizontal line.

AUTHOR:

Jean-Francois VIBERT and Sylvain Hanneton

CHU Saint-Antoine CNRS UA1162 Universite Paris VI et Ecole Normale Superieure

SEE ALSO: PS_CIRCLE, PS_DISK, PS_ELLIPSE, PS_ELLIP_DISK

17 - PS_CIRCLE

NAME: PS_CIRCLE Draws an empty circle

SYNOPSIS:

```
#include <plot_ps.h>

ps_circle(r)
float   r;   /* radius always in cms */
```

DESCRIPTION:

PS_CIRCLE draws an empty circle whose the center is located at the current position. It comes back at the same current position.

AUTHOR:

Jean-Francois VIBERT et Sylvain Hanneton

CHU Saint-Antoine CNRS UA1162 Universite Paris VI et Ecole Normale Superieure

SEE ALSO:

PLOT1 et PLOT2

18 - PS_DISK

NAME: PS_DISK Draws a filled disk

SYNOPSIS:

```
#include <plot_ps.h>

ps_disk(r,gray)
float   r,   /* radius always in cms */
        gray;/* shading 0-1 */
```

DESCRIPTION:

PS_DISK draws an filled disk whose the center is located at the current position. It comes back at the same current position. The shading is between 0 (black) and 1 (white).

AUTHOR:

Jean-Francois VIBERT et Sylvain Hanneton

CHU Saint-Antoine CNRS UA1162 Universite Paris VI et Ecole Normale Superieure

SEE ALSO:

PLOT1 et PLOT2

19 - PS_ELLIPSE

NAME: PS_ELLIPSE Draws an empty ellipse

SYNOPSIS:

```
#include <plot_ps.h>

ps_ellipse(h_d,v_d)
float   h_d,    /* horizontal diameter in cms */
        v_d;    /* vertical diameter in cms */
```

DESCRIPTION:

PS_ELLIPSE draws an empty ellipse whose the center is located at the current position. The parameters are the horizontal and vertical diameters (not radius) in cms. It comes back at the same current position.

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

20 - PS_ELLIP_DISK

NAME: PS_ELLIP_DISK Draws a filled ellipse

SYNOPSIS:

```
#include <plot_ps.h>

ps_ellip_disk(h_d,v_d,gray)
float   h_d,    /* horizontal diameter in cms */
        v_d,    /* vertical diameter in cms */
        gray;   /* shading 0-1 */
```

DESCRIPTION:

PS_ELLIP_DISK draws a filled ellipse whose the center is located at the current position. The parameters are the horizontal and vertical diameters (not radius) in cms. It comes back at the same current position. The shading is between 0 (black) and 1 (white).

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

21 - PS_MOVEMENTS

NAME: PS_MOVEMENTS Functions to move pen and to draw

SYNOPSIS:

```
#include <plot_ps.h>

ps_user()

ps_cms()

ps_mover(x,y)
float x, y;

ps_movea(x,y)
float x, y;

ps_drawr(x,y)
float x, y;

ps_drawa(x,y)
float x, y;

ps_home()
```

DESCRIPTION:

With `ps_user()` you go in user space,the coordinates are now expressed in user units and not in centimeters. .

With `ps_cms()` you go in paper space,the coordinates are now expressed in centimeters and not in user units.

With the following functiuns,you are able to draw in the current space: ps_move move without drawing ,ps_draw move the pen but draws too .The coordinates are absolutes (a suffixe) or relatives (r suffixe).

`ps_movea(x,y)` moves to the absolute (x,y) point.

`ps_mover(x,y)` moves to the relative (x,y) point.

`ps_drawa(x,y)` draws a line to the absolute (x,y) point.

`ps_drawr(x,y)` draws a line to the relative (x,y) point.

`ps_home()` goes back to the origin.

Dr Jean-Francois Vibert

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

plot[0-3]_ps,

22 - ps_3Dsetup

NAME: ps_3Dsetup Set up a 3D reference

SYNOPSIS:

```
#include <plot_ps.h>
ps_3Dsetup(taillex,tailley,taillez,
        minx,maxx,miny,maxy,minz,maxz,
        angle,phi,dist,dist2);
float taillex,tailley,taillez,       /* axes lenghts in cm*/
      minx,maxx,miny,maxy,minz,maxz, /* minima and maxima
                                        of user values */
      angle,phi,                     /* angular coord of 3D ref */
      dist,      /* distance reference-screen */
      dist2;      /* distance eye-screen */
```

DESCRIPTION:

ps_3Dsetup must be call at the very begining of a 3D drawing.

The reference is oriented as the following if the angles are 0:

- y axe is upward
- x axe is rightward
- z axe is frontward (toward the observer)

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

PLOT1 et PLOT2,p3Dbar,p3Dframe,p3Dbox,p3Dcalcul...

23 - ps_3Dcalcul

NAME: ps_3Dcalcul Computes projection and rotation from 3D to 2D space

SYNOPSIS:

```
#include <plot_ps.h>

ps_3Dcalcul(y1,z1,x1,x,y)
float    x1,y1,z1,    /* 3D coordinates */
        *x,*y;        /* 2D coordinates */
```

DESCRIPTION:

ps_3Dcalcul translates the coordinates from the 3D reference (drawn by p3Daxes for example) to a 2D reference x,y of the corresponding projection on the screen.

The reference is oriented as the following if the angles are 0:

- y axe is upward
- x axe is rightward
- z axe is frontward (toward the observer)

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

PLOT1 et PLOT2,ps_3Dbar,ps_3Dframe,ps_3Dbox,ps_3Daxes...

24 - ps_3Daxes

NAME: ps_3Daxes Draws a 3D reference

SYNOPSIS:

```
#include <plot_ps.h>
ps_3Daxes(boite,outline);
int   boite,
      outline;     /* true if a box is needed */
```

DESCRIPTION:

ps_3Daxes draws a reference triaxes in 3D whoses axes have not tick marks. If the BOITE flag is TRUE, then, an open box is drawn in the background in order to add depth feeling to the drawing. If the out line flag is TRUE, the sides of the box are outlined.

The reference is oriented as the following if the angles are 0:

- y axe is upward
- x axe is rightward
- z axe is frontward (toward the observer)

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

PLOT1 et PLOT2,ps_3Dbar,ps_3Dframe,ps_3Dbox,ps_3Dcalcul...

25 - ps_3Dbox

NAME: ps_3Dbox Draws a box in 3D (only 3 faces are represented)

SYNOPSIS:

```
#include <plot_ps.h>

ps_3Dbox(k,l,m,la,ep,lo,outline)
float   k,l,m,     /* coordinates of the origin point of the box */
        la,ep,lo;  /* dimensions of the box */
\tmpind{-1.0cm}

int     outline;   /* outline flag */
```

DESCRIPTION:

ps_3Dbox draws the back sides of a box in a way to always get visible the inside of the box. The given coordinates are those of the bottom, left and back corner. Then are width, height and thickness of the box. Hidden surfaces are correctly processed. It draws a box alone, without axes, and the box can be drawn anywhere, at the difference of ps_3Daxes. If the flag `outline` is true, the visible sides of the box are outlined.

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

PLOT1 et PLOT2,ps_3Dbar,ps_3Dframe,ps_3Dbox,ps_3Dcalcul...

26 - ps_3Dbar

NAME: ps_3Dbar Draws a bar in 3D

SYNOPSIS:

```
#include <plot_ps.h>

ps_3Dbar(k,l,m,la,ep,lo,outline)
float   k,l,m,     /* coordinates of the origin point of the bar */
        la,ep,lo;  /* dimensions of the bar */
\tmpind{-1.0cm}

int     outline ;  /* outline flag */
```

DESCRIPTION:

ps_3Dbar draws the front sides of a bar in a way to always get visible the outside of the bar. The given coordinates are those of the bottom, left and back corner. Then are width, height and thickness of the bar. Hidden surfaces are correctly processed. It draws a bar alone, without axes, and the bar can be drawn anywhere. If the `outline` flag is true, the sides of the bar are outlined.

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI ENS

SEE ALSO: PLOT1 et PLOT2,ps_3Daxes,ps_3Dframe,ps_3Dbox,ps_3Dcalcul...

27 - ps_3Dframe

NAME: ps_3Dframe Draws a bar skeleton in 3D

SYNOPSIS:

```
#include <plot_ps.h>

ps_3Dframe(k,l,m,la,ep,lo)
float   k,l,m,     /* coordinates of the origin point of the bar */
        la,ep,lo;  /* dimensions of the box */
```

DESCRIPTION:

ps_3Dframe draws the skeleton of a bar. The given coordinates are those of the bottom, left and back corner. Then are width, height and thickness of the bar. It draws a bar alone, without axes, and the bar frame can be drawn anywhere.

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

PLOT1 et PLOT2,ps_3Dbar,ps_3Dframe,ps_3Dbox,ps_3Dcalcul...

28 - ps_3Dclosebox

NAME: ps_3Dclosebox Close a box with skeleton in 3D

SYNOPSIS:

```
#include <plot_ps.h>

ps_3Dclosebox(k,l,m,la,ep,lo)
float   k,l,m,     /* coordinates of the origin point of the bar */
        la,ep,lo;  /* dimensions of the box */
```

DESCRIPTION:

ps_3Dclosebox closes a box (drawn with ps_3Dbox for example) with the corresponding front skeleton. The given coordinates are those of the bottom, left and back corner. Then are width, height and thickness of the box. It allows to enclose items in this box. With this function you can also outline the front sides of a bar drawn with ps_3Dbar for example.

ps_3Dclosebox must be invoked after all the objects enclosed in the box are drawn.

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

PLOT1 et PLOT2,ps_3Dbar,ps_3Dframe,ps_3Dbox,ps_3Dcalcul...

29 - PS_POLYLINE

NAME: PS_POLYLINE Draws a polyline

SYNOPSIS:

```
#include <plot_ps.h>

ps_polyline(x,y,nb_pts,dot,dot_type,dot_size)
float   x[],        /* array with x coordinates */
        x[];        /* array with y coordinates */
int     nb_pts,     /* nb of values in x (and y) */
        dot,        /* TRUE if dots must be drawn */
        dot_type;   /* dot type from a list of 16 */
float   dot_size;   /* dot size in cms */
```

DESCRIPTION:

PS_POLYLINE() draws a polygonal line whose absolute coordinates are in 2 arrays x and y, beginning at the current position. If `dot` is TRUE, a dot is drawn at each articulation of the polyline (included extremities). The dot size is given in cms by `dot_size`. The `dot_type` must be choosen among the following values:

value 1 and 2: square empty and filled
value 3 and 4: losange empty and filled
value 5 and 6: up triangle empty and filled
value 7 and 8: down triangle empty and filled
value 9 and 10: circle empty and filled
value 11 and 12: left triangle empty and filled
value 12 and 13: right triangle empty and filled
value 14: x (cross)
value 15: + (plus sign)
serie 16: . (dot)

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

30 - PS_CLOSED_CURVE

NAME: PS_CLOSED_CURVE Draws a possibly filled surface

SYNOPSIS:

```
#include <plot_ps.h>

ps_closed_curve(x,y,nb_pts,contour,fill,gray)
float   x[],        /* array with x coordinates */
        x[];        /* array with y coordinates */
int     nb_pts,     /* nb of values in x (and y) */
        contour,    /* t or f line around or not */
        fill;       /* t or f filled or not */
float   gray;       /* gray filling color */
```

DESCRIPTION:

PS_CLOSED_CURVE() draws polygonal surface whose absolute coordinates are in 2 arrays x and y, beginning at the current position. The curve will be close automatically to the first point. It is possible to decide to draw a contour around the surface. This surface can be filled in gray, whose color is given by gray.

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

31 - PS_BEZIER

NAME: PS_BEZIER Draws a Bezier curve

SYNOPSIS:

```
#include <plot_ps.h>

ps_bezier(x,y,dot,dot_type,dot_size)
float   x[],        /* array with x coordinates */
        x[];        /* array with y coordinates */
int     dot,        /* TRUE if dots must be drawn */
        dot_type;   /* dot type from a list of 16 */
float   dot_size;   /* dot size in cms */
```

DESCRIPTION:

PS_BEZIER() draws a Bezier cubic curve whose the 3 absolute coordinates are in 2 arrays x and y, beginning at the current position.

WARNING: the ps_bezier() draws only a small bezier curve on four (4) control points: the current position and the 3 control points in the x and y arrays.

If `dot` is TRUE, a dot is drawn at each Bezier cubic control point (included extremities). The dot size is given in cms by `dot_size`. The dot size is given in cms by `dot_size`. The `dot_type` must be choosen among the following values:

value 1 and 2: square empty and filled
value 3 and 4: losange empty and filled
value 5 and 6: up triangle empty and filled
value 7 and 8: down triangle empty and filled
value 9 and 10: circle empty and filled
value 11 and 12: left triangle empty and filled
value 12 and 13: right triangle empty and filled
value 14: x (cross)
value 15: + (plus sign)
serie 16: . (dot)

AUTHOR:

Jean-Francois VIBERT et Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI et Ecole Normale Superieur

SEE ALSO:

PLOT1 et PLOT2

32 - PS_POINT

NAME: PS_POINT Draws a point centered on the current position

SYNOPSIS:

```
#include <plot_ps.h>

ps_point(dot_type,dot_size)
int     dot_type;         /* type of the dot (1-16) */
float   dot_size;             /* height of the symbol in cms */
```

DESCRIPTION:

PS_POINT draws a point centered at the current position, whose height is given by dot_size (in cms). 16 different types are available:

value 1 and 2: square empty and filled
value 3 and 4: losange empty and filled
value 5 and 6: up triangle empty and filled
value 7 and 8: down triangle empty and filled
value 9 and 10: circle empty and filled
value 11 and 12: left triangle empty and filled
value 12 and 13: right triangle empty and filled
value 14: x (cross)
value 15: + (plus sign)
serie 16: . (dot)

AUTHOR:

Jean-Francois VIBERT

CHU Saint-Antoine CNRS UA1162 Universite Paris VI

SEE ALSO:

PLOT1 et PLOT2

33 - PS_FRAME

NAME: PS_FRAME Draws a frame at the current position

SYNOPSIS:

```
#include <plot_ps.h>

ps_frame(width,height)
float   width,        /* of the rectangle */
        height;       /* of the rectangle */
```

DESCRIPTION:

PS_FRAME draws an empty transparent frame whose position of the lower-left angle is at the current position.

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

PLOT1 and PLOT2

34 - PS_WRITE

NAME: PS_WRITE Writes a character string or a number

SYNOPSIS:

```
#include <plot_ps.h>

ps_writeah(x,y,string,taille)
float   x,      /* X coordinate     */
        y,      /* X coordinate     */
        taille; /* character height */
char    *string;/* string to write  */


ps_writeav(x,y,string,taille)
float   x,      /* X coordinate     */
        y,      /* X coordinate     */
        taille; /* character height */
char    *string;/* string to write  */

ps_writefloatav(x,y,number,taille)
float   x,      /* X coordinate     */
        y,      /* X coordinate     */
        number, /* number to write  */
        taille; /* character height */

ps_writefloatah(x,y,number,taille)
float   x,      /* X coordinate     */
        y,      /* X coordinate     */
        number, /* number to write  */
        taille; /* character height */

ps_writeintav(x,y,number,taille)
float   x,      /* X coordinate     */
        y,      /* X coordinate     */
        taille; /* character height */
int     number; /* number to write  */

ps_writeintah(x,y,number,taille)
float   x,      /* X coordinate     */
        y,      /* X coordinate     */
        taille; /* character height */
int     number; /* number to write  */
```

DESCRIPTION:

PS_WRITE..A..() with the following code:

nothing -> string
int -> integer
float -> float
h -> horizontal
v -> vertical

`ps_writeah` writes a character string horizontally, with x and y as absolute positions.

`ps_writeav` writes a character string vertically, with x and y as absolute positions

`ps_writefloatah` writes a float horizontally, with x and y as absolute positions

`ps_writefloatav` writes a float vertically, with x and y as absolute positions

`ps_writeintah` Writes an integer horizontally, with x and y as absolute positions

`ps_writeintav` Writes an integer vertically, with x and y as absolute positions

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

PLOT1 and PLOT2

35 - PS_AXES

NAME: PS_AXES Draws a two axes centered on the extremal values

SYNOPSIS:

```
#include <plot_ps.h>
ps_axe(taillex,tailley,minx,maxx,miny,maxy,legende_x,legende_y,cadre)
float taillex,tailley,      /* axes lenghts in cm */
      minx,maxx,miny,maxy;  /* maximal values */
char  *legende_x,           /* X axis legend */
      *legende_y;           /* Y axis legend */
int   cadre;                /* flag */
```

DESCRIPTION:

PS_AXES draws two axes centered on the maximal and minimal values of the axes(without tick marks).The absolute values of the maxima and minima are written at the bottom of the axes.If the flag is TRUE, a frame is drawn around the axes.

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

PLOT1 and PLOT2

36 - PS_INCLUDE

NAME: PS_INCLUDE Includes a PostScript file

SYNOPSIS:

```
#include <plot_ps.h>

ps_include(file_name)
char     *file_name; /* name of the PS file */
```

DESCRIPTION:

PS_INCLUDE includes a PostScript file in the currently opened PS file. It is the responsability of the user to be sure of the PS content of the file, and of its coherence (scale, units, position, page number, etc...).

PS_INCLUDE returns TRUE if the file exits and FALSE if a problem was encountered during the file opening.

AUTHOR:

Jean-Francois VIBERT et Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI et Ecole Normale Superieur

SEE ALSO:

PLOT1 et PLOT2

37 - PS_CODE

NAME: PS_CODE Includes PostScript code

SYNOPSIS:

```
#include <plot_ps.h>

ps_code(ps_code)
char     *ps_code; /* name of the PS file */
```

DESCRIPTION:

PS_CODE includes PostScript code in the currently opened PS file. It is the responsability of the user to be sure of the PS content of the included code, and of its coherence (scale, units, position, page number, etc...).

AUTHOR:

Jean-Francois VIBERT et Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI et Ecole Normale Superieur

SEE ALSO:

PLOT1 et PLOT2

38 - PS_TICKED_AXES

NAME: PS_TICKED_AXES Draws a ticked reference

SYNOPSIS:

```
#include <plot_ps.h>
ps_ticked_axes(tx,ty,mnx,mxx,mny,mxy,legendx,legendy,hy,cadre)
float   tx,ty,   /* length of the two axes */
        mnx,mxx, /* minima and maxima in your units */
        mny,mxy,
        hy;      /* length of the smallest ticked mark in cms */
char    legendx[60], /* legend on the horizontal axe */
legendy[60]; /* legend on the vertical axe   */
int     cadre; /* flag = 1 if you want a frame around the axe */

ps_X_axis(tx,mnx,mxx,legendx,hy)
float   tx,      /* length of the X axis */
mnx,mxx, /* minima and maxima in your units */
hy;      /* length of the smallest ticked mark in cms */
char  legendx[60]; /* legend on the horizontal axe */

ps_Y_axis(ty,mny,mxy,legendy,hy)
float   ty,      /* length of the Y axis */
mny,mxy, /* minima and maxima in your units */
hy;      /* length of the smallest ticked mark in cms */
char  legendy[60]; /* legend on the horizontal axe */
```

DESCRIPTION:

PS_TICKED_AXES draws a reference width ticked axes. The ticked marks are well computed. If the flag CADRE is true a frame is drawn around the reference.

PS_X_AXIS and PS_Y_AXIS do the same things but in one direction only, x or y..

To choose the font of the legends, before use the ps_font instruction.

The minima and maxima are in the user space units.

AUTHOR:

Sylvain Hanneton

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO:

39 - PS_HISTO

NAME: PS_HISTO Draws an histogramm

SYNOPSIS:

```
#include <plot_ps.h>

ps_histo(popu,tx,ty,gap,nb_class,flag)
int   popu[MAX_DATA], /* array of values to be drawn */
      nb_class,           /* number of classes */
      flag; /* 0 no calcul 1 vertical 2 horizontal other = all */
float tx,ty, /* length of the axes */
      gap;   /* width of a class in user units */
```

DESCRIPTION:

PS_HISTO draws the histogramm of the given array (integer values). The flag FLAG indicates if you want or not indicate statistical reference on the drawing. In this case, the mathematic library must be loaded when linking the task.

If the flag is 0 no indications are given. If the flag is 1 the mean value of the histogramm is indicated with a horizontal gray line. If the flag is 2 then the expected value is indicated with a vertical line. If flag takes another value, both are indicated.

AUTHOR:

Sylvain HANNETON

CHU Saint-Antoine CNRS UA1162 Universite Paris VI & Ecole Normale Superieure

SEE ALSO: