BNL-105200-2014-TECH

Booster Technical Note No. 156;BNL-105200-2014-IR

CPLOT: AN APOLLO PLOTTING PROGRAM USING CALCOMP AND GPR

A. U. Luccio

January 1990

Collider Accelerator Department

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC)

DISCLAIMER

CPLOT: AN APOLLO PLOTTING PROGRAM

USING CALCOMP AND GPR

AGS
BOOSTER TECHNICAL NOTE
NO. 156

A. LUCCIO

JANUARY 9, 1990

ACCELERATOR DEVELOPMENT DEPARTMENT
BROOKHAVEN NATIONAL LABORATORY
UPTON, NEW YORK 11973

CPLOT: AN APOLLO PLOTTING PROGRAM USING CALCOMP AND GPR

A.LUCCIO

January 1990

1. Introduction

"cplot", a Fortran plotting program using "calcomp" and "gpr" has been developed to plot on the Apollo workstations line diagrams and any pattern of circles, rectangles, ellipses, etc. The program is currently being used to check the algorithms for the model based control of the AGS Booster.

calcomp is an old graphic package by California Computer Products, Inc., edited by R.G.Wittlock at Brookhaven[1]. gpr is one of graphic packages available on the Apollo workstations[2]. Agnes Abola of the AGS Control Group has created a user library on the AGS Apollo network and has put together calcomp and gpr, to form the basis on which cplot operates.

cplot is by no means a final product, but we find it already very useful as is.

---

[1] Wittlock, R.G. "CALCOMP Language Reference Manual", Report BNL 23987, 1978

[2] "Domain Graphics Primitive Resource Call Reference", Apollo Computers Inc., Order No. 007194, Rev 02, 1987
also: "Programming with Domain Graphics Primitives", Order No. 005808, Rev 01, 1987

2. Compile and bind cplot

The fortran-77 source for cplot is in Appendix A. cplot is compiled with the Domain fortran compiler. At the Aegis[3] prompt $, type:

$ftn cplot

then, bind with calcomp and gpr as follows:

$bind cplot.bin -inlib [dir]/controls.userlib -inlib [dir]/calcomp.gprlib -b cplot

where (at the present time. If it does not work, ask Agnes):

[dir] = /users/operations/com

[3]Aegis is an Apollo shell command language. See: "Domain System Call Reference", Apollo Computers Inc., Order No. 002547, Rev. 04, 1987.

3. Use of cplot. Parameter file

The input is the file "cplot.in" (App. B) containing the following:

temp parameter file

xx)
yy
data files (files containing the data to plot)
....
zz)

where temp,xx, yy .... zz are any file names

The plot of Fig. 1 (a collection of distorted orbits in the Booster generated by MAD[4] for several random extractions of position and field errors in the magnets) uses the parameter file shown in App. C. This file is a namelist "plotp" input for cplot, containing the following:

title	(will appear on the top of the plot);
xx and yy	dimensions of the plot in screen inches;
xlabel and ylabel	axis labels;
xpmin, xpmax, ypmin, ypmax	minima and maxima of the values we want on each axis, in real coordinates;
isym	in calcomp, Ref[1] , page 3-27, a symbol code for the data point to draw. In the example, isym = 3 means a cross or a "+";
n_data_max	for a curve - self explanatory - to keep low the dimensions of the program;
max_plot	as above;
n_y	how many curves to plot from the same data file;
window	Apollo graphics window, in pixels. In the order: top edge from screen top, left edge from screen left, x-size, y-size.

[4] Iselin, F. Ch, and Niederer, J., "The MAD Program (Methodical Accelerator Design, Version 6, User's Reference Manual", Rept. CERN/LEP-TH/87-33, Geneva, April 1987

Notes: (i) Each label, like the title and the axis labels must be terminated with a $ sign.
(ii) The axis are automatically numbered. To obtain nice numbers, xmin-xmax should be a nice multiple of xx. Likewise for ymax-ymin and yy.

For a good use of cplot you want to put bookkeeping informations in the title, to be printed on the graph. For this, the editing Aegis script "ctime" is used. ctime inserts date, time, run id and page # after the title, changing the parameter file into "temp".

Therefore, after cplot.in is prepared, to run the program, first type at the Aegis prompt $:

$ctime [parameter_file] temp [run#] [page #]

then invoke cplot:

$cplot

4. Data files. Data type.

A data file for cplot must have a keyletter representing a data type in row 1, column 1, to specify how to interpret the data which follow, and the keyletter "Q" after the last row of data, column 1. cplot ignores what is written in the remainder of the first row and after the Q. A file may contain many blocks of data, each initiated by a proper keyletter.

Keyletters are:

L

Data are for a solid line plot. The first column is x, the following columns are sets of y. cplot plots n_ y curves of y vs. x. Example in App. D. At the present, only linear coordinate axis are implemented.

D

Dashed line plot.

X

Scatter plot of the y's vs. x. The symbol used is determined by isym.

R

Draw rectangles. Following calcomp, the data in each line are: the coords of the left lower corner, the x and y dimensions and the rotation angle in degrees. App. E shows the file to plot the Booster layout of Fig. 1.

C

Draw arcs. Data in each line are: the coords of the center, start and end radii (it draws also spirals), start and end angles. Example in App. F, a data file for Fig. 2.

P

Draws straight segments. In each line: start coords, length, angle (in deg) (App. D). If the lines are too long, the routine "inside" clips them, so they stay inside the frame. This is useful if we want to mess up with the window in the parameter file for zooming the plot.

T

Draws text anywhere on the screen. For each test string, a first line gives: the coordinates of the left lower corner of the first character, the height of the characters and the angle of the text string, and a second line gives the string (terminated by a $). Example in App. G, a data file for Fig. 1.

E

Ellipses. A line represents: coords of the center, x and y axis, angle of rotation (in rad). The phase space ellipses in Fig. 3 (proton injection in the Booster) are obtained with this option.

Appendix A . Fortran code for "cplot".

```
*************************************************************************
*                          CPLOT.FTN                                    *
* //acn40d01/reality/luccio/graf                                       *
* A.Luccio BNL Sept-1988                                               *
* using CALCOMP and GPR                                                *
* Modif:  Thursday, November 16, 1989   3:57:12 pm (EST)               *
*************************************************************************
      program cplot

c%include '/sys/ins/base.ins.ftn'
c%include '/sys/ins/gpr.ins.ftn'

      integer          nzch(10)
      integer*2        stream_id,window(4),bitm_size(2)
     &                 ,b_scale(2),time(3)
      character        yorn,data_type
      character*64     xlabel,ylabel,zlabel(10),title
      character*128    infile,ppar_file,xi_fname,anychar
      logical          start,spline

      namelist    /plotp/
     &             xx,yy,xlabel,ylabel,zlabel,title,
     &             xpmin,xpmax,ypmin,ypmax,isym,
     &             n_data_max,max_plot,n_y,window,
     &             spline,interac

      common        /plotp/
     &              xx,yy,dxp,dyp,b_scale,
     &              xpmin,xpmax,ypmin,ypmax,isym,
     &              xlabel,ylabel,zlabel,title,
     &              nxch,nych,nzch,ntch,
     &              nv,n_y,n_data_max,max_plot,spline

      data     n_data_max,max_plot,window/10000,512,100,500,750,750/
      data     stream_id/-1/
     &         ,start,spline/.true.,.false./

      open     (19,file='cplot.in')

*      write    (*,6000)
6000  format   ('Enter xxx.ppar file: ',$)
*      read     (*,'(a128)') ppar_file
      read     (19,'(a128)') ppar_file
*      write    (*,'(a128)') ppar_file
      open     (14,file=ppar_file)
      read     (14,plotp)
      close    (14)

        dxp    = (xpmax-xpmin)/xx
        dyp    = (ypmax-ypmin)/yy
        x_t    = 0.1*(xpmax-xpmin)/dxp
        h_t    = (ypmax-ypmin)/40/dyp
        y_t    = 1.025*(ypmax-ypmin)/dyp
*        bitm_size(1) = window(3)
*        bitm_size(2) = window(4)
c        b_scale(1)   = bitm_size(1)/xx
c        b_scale(2)   = bitm_size(2)/yy

      do i = 1,64
          if (xlabel(i:i) .eq. '$')    nxch = i-1
          if (ylabel(i:i) .eq. '$')    nych = i-1
          if (title(i:i)  .eq. '$')    ntch = i-1
        do j = 1,10
          if (zlabel(j)(i:i) .eq. '$') nzch(j) = i-1
        end do
      end do
c     ------------------------------------------------------
100   continue
*      write    (*,6010)
6010  format   ('Enter xxx.pdat file: ',$)
*      read     (*,'(a128)') infile
```

Appendix A . Fortran code for "cplot"(continue).

```
      read    (19,'(a128)',end=900) infile
*      write   (*,'(a128)') infile
*      if (infile .eq. 'end') goto 900

      if (start) then
        call new_initialize_gpr (window)
        call calinit
        call erase
        call acquire
        call rect   (0.,0.,yy,xx,0.,isym)
        call symbol (x_t,y_t,h_t,title,0.,ntch)
        call release
      endif

c        call gpr_$init
c     &(gpr_$borrow,int2(1),bitm_size,int2(1),int4(1),status)

      open    (10,file=infile)
200   read    (10,'(a)') data_type
*      print   *,'(1)data_type= ',data_type
        call acquire
      if (data_type.eq.'L' .or.
     &    data_type.eq.'X' .or.
     &    data_type.eq. 'D')
     &  call line_plot   (start,data_type,*100,*200)
        call layout_plot (start,data_type,*100,*200)

900   continue
*      pause
      close   (19)
      write   (*,6020)
6020  format  ('Type any char to exit plot: ',$)
      read    (*,'(a128)') anychar

      call erase
      call terminate_gpr

      end
c-----------------------------------------------------------------
c-----------------------------------------------------------------
      subroutine line_plot (start,data_type,*,*)

      parameter     (m=1002)
      integer       nzch(10)
      integer*2     l_scale,b_scale(2),bx(m),by(m)
      integer*4     status
      real          x(m),y(m),z(5,m)
      character     data_type
      character*64  xlabel,ylabel,zlabel(10),title
      logical       start,spline

       common       /plotp/
     &              xx,yy,dxp,dyp,b_scale,
     &              xpmin,xpmax,ypmin,ypmax,isym,
     &              xlabel,ylabel,zlabel,title,
     &              nxch,nych,nzch,ntch,
     &              nv,n_y,n_data_max,max_plot,spline

                              lintyp = 0
      if (data_type .eq. 'X') lintyp = -1

        i    = 1
100     read   (10,*,err=104) x(i),(z(k,i), k=1,n_y)
          if (x(i).ge.xpmax) xpmax = x(i)
          if (x(i).le.xpmin) xpmin = x(i)
        do k = 1,n_y
          if (z(k,i).ge.ypmax) ypmax = z(k,i)
          if (z(k,i).le.ypmin) ypmin = z(k,i)
        end do
          i     = i +1
          if (i .gt. n_data_max) goto 104
```

Appendix A . Fortran code for "cplot"(continue).

```
          goto 100
104     nv       = i-1
        n_skip   = nv/max_plot +1
        j        = 0
      do 110 i   = 1,nv,n_skip
        j        = j +1
        x(j)     = x(i)
c        bx(j)    = int2(b_scale(1)*(x(j)-xpmin))
      do 110 k   = 1,n_y
110     z(k,j)   = z(k,i)
        nv       = j
        x(nv+1)  = xpmin
        x(nv+2)  = (xpmax-xpmin)/xx


      do 120    k  = 1,n_y
          l_scale = int2(1 +10*(k-i))
          call gpr_$set_linestyle (gpr_$dotted,l_scale,status)
        do       i = 1,nv
          y(i)     = z(k,i)
c          by(i)    = int2(yy-b_scale(2)*(y(i)-ypmin))
c          print *,k,i,bx(i),by(i)
        end do
          y(nv+1) = ypmin
          y(nv+2) = (ypmax-ypmin)/yy
          call rescale
        if (start) then
          call axis  (0.,0.,xlabel,-nxch,xx,0.,x(nv+1),x(nv+2))
          call axis  (0.,0.,ylabel,nych,yy,90.,y(nv+1),y(nv+2))
          start   = .false.
        endif
          npts    = nv
          if (spline) npts = -nv
*          call fline (x,y,npts,1,lintyp,isym)
*          if ( (k/2).eq.(k/2.) .or.
          if (
     &         data_type .eq. 'D' ) then
            call dashl (x,y,npts,1,lintyp,isym)
          else
            call line (x,y,npts,1,lintyp,isym)
          endif
c          call gpr_$move (bx(1),by(1),status)
c          call gpr_$polyline (bx,by,int2(nv),status)
120   continue

      call release

      backspace   (10)
      read        (10,'(a)') data_type
      if (data_type .eq. 'Q') then
*        print *,'(2)data_type= ',data_type
        close     (10)
        return    1
      else
        backspace (10)
        return 2
      endif

      end
c-----------------------------------------------------------------
c-----------------------------------------------------------------
      subroutine layout_plot (start,data_type,*,*)

      integer       nzch(10)
      integer*2     b_scale(2)
      character     data_type
      character*64  xlabel,ylabel,zlabel(10),title,text
      logical       start,spline

       common       /plotp/
     &              xx,yy,dxp,dyp,b_scale,
```

Appendix A . Fortran code for "cplot"(continue).

```
     &              xpmin,xpmax,ypmin,ypmax,isym,
     &              xlabel,ylabel,zlabel,title,
     &              nxch,nych,nzch,ntch,
     &              nv,n_y,n_data_max,max_plot,spline

      if (start) then
        call axis  (0.,0.,xlabel,-nxch,xx,0.,xpmin,dxp)
        call axis  (0.,0.,ylabel,nych,yy,90.,ypmin,dyp)
        start  = .false.
      endif

      call rescale

      if (data_type .eq. 'R') goto 200
      if (data_type .eq. 'C') goto 300
      if (data_type .eq. 'P') goto 400
      if (data_type .eq. 'T') goto 500
      if (data_type .eq. 'E') goto 600
      if (data_type .eq. 'Q') goto 900
c     ------------------------------------------------------ R: Rectangle
200   read  (10,*,err=900) xrr,yrr,w,h,angle
        x_r       = (xrr -xpmin)/dxp
        y_r       = (yrr -ypmin)/dyp
        hei_r     = h/dyp
        wid_r     = w/dxp
        ang_r     = angle
        call rect (x_r,y_r,hei_r,wid_r,ang_r,3)
      goto 200
c     ------------------------------------------------------ C: Circle
300   read  (10,*,err=900) xcc,ycc,rcc1,rcc2,angle1,angle2
        x_c       = (xcc -xpmin)/dxp
        y_c       = (ycc -ypmin)/dyp
        rad_c1    = rcc1/dyp
        rad_c2    = rcc2/dyp
        ang_c1    = angle1
        ang_c2    = angle2
        call circl (x_c,y_c,ang_c1,ang_c2,rad_c1,rad_c2,0.5)
      goto 300
c     ------------------------------------------------------ P: Polygon
400   read  (10,*,err=900) xll,yll,sll,angle
        call inside (xll,yll,sll,angle,data_type)
        if (sll .eq. 0.) goto 400
        i_l       = i_l +1
        x_l       = (xll -xpmin)/dxp
        y_l       = (yll -ypmin)/dyp
        s_len     = sll/dyp
        ang_l     = angle
        call poly (x_l,y_l,s_len,1.,ang_l)
      goto 400
c     ------------------------------------------------------ T: Text
500   read  (10,*,err=900) xtt,ytt,htt,angle
*      print *,'xt,yt,ht,angle= ',xtt,ytt,htt,angle
      read  (10,'(a64)') text
*      print *,'text=',text
        i_t       = i_t +1
        x_t       = (xtt -xpmin)/dxp
        y_t       = (ytt -ypmin)/dyp
        hei_t     = htt/dyp
        ang_t     = angle
        do i = 1,64
          if (text(i:i) .eq. '$') n_t = i-1
        end do
        call symbol (x_t,y_t,hei_t,text,ang_t,n_t)
      goto 500
c     ------------------------------------------------------ E: Ellipse
600   read  (10,*,err=900) xc,yc,a,b,theta
        aa        = a*sqrt((cos(theta)/dxp)**2 +(sin(theta)/dyp)**2)
        bb        = b*sqrt((sin(theta)/dxp)**2 +(cos(theta)/dyp)**2)
        xxc       = (xc-xpmin)/dxp
        yyc       = (yc-ypmin)/dyp
        angle     = atan2((dxp*tan(theta)),dyp)
```

Appendix A , Fortran code for "cplot"(continue).

```
        dega      = angle*180/3.141592654
        xpage     = xxc +aa*cos(angle)
        ypage     = yyc +aa*sin(angle)
        call elips (xpage,ypage,aa,bb,dega,0.,360.,3)
      goto 600
c     ------------------------------------------------------ P: Polygon
900   call release
      backspace   (10)
      read        (10,'(a)') data_type
      if (data_type .eq. 'Q') then
*       print *,'(3)data_type= ',data_type
        close     (10)
        return    1
      else
        backspace (10)
        return    2
      endif

      end
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      subroutine inside (xl,yl,sl,angle,data_type)

      integer       nzch(10)
      integer*2     b_scale(2)
      dimension     x(6),y(6),xc(2),yc(2)
      character*64  xlabel,ylabel,zlabel(10)
      character     data_type
      logical       spline

      common        /plotp/
     &              xx,yy,dxp,dyp,b_scale,
     &              xpmin,xpmax,ypmin,ypmax,isym,
     &              xlabel,ylabel,zlabel,title,
     &              nxch,nych,nzch,ntch,
     &              nv,n_y,n_data_max,max_plot,spline

      data         pie/3.14159265/

      if (data_type .eq. 'P') goto 100

100     theta   = angle*pie/180

      do 110 i  = 1,5,2
        x(i)    = xl
        x(i+1)  = xl +sl*cos(theta)
        y(i)    = yl
110     y(i+1)  = yl +sl*sin(theta)

      if (x(1) .eq. x(2)) goto 130
        x(3)    = xpmin
        x(4)    = xpmax
      do 120 i  = 3,4
120     y(i)    = y(1) +(x(i)-x(1))*tan(theta)
130   if (y(1) .eq. y(2)) goto 150
        y(5)    = ypmin
        y(6)    = ypmax
      do 140 i  = 5,6
140     x(i)    = x(1) +(y(i)-y(1))/tan(theta)
150       xold  = 9999
          yold  = 9999
          j     = 1
      do 160 i  = 1,6
        if (j .gt. 2) goto 170
        if (   (x(i)-xpmax)*(x(i)-xpmin) .le. 0
     &    .and. (y(i)-ypmax)*(y(i)-ypmin) .le. 0
     &    .and. (x(i).ne.xold .or. y(i).ne.yold)) then
          xc(j) = x(i)
          yc(j) = y(i)
          xold  = x(i)
          yold  = y(i)
```

Appendix A . Fortran code for "cplot"(continue).

```
          j     = j +1
        endif
160   continue

170     xl      = xc(1)
        yl      = yc(1)
        sl      = sqrt((xc(2)-xc(1))**2 +(yc(2)-yc(1))**2)
      if (j .le. 2) sl = 0
        angle   = 90
      if (xc(2).ne.xc(1)) then
        theta   = atan2( (yc(2)-yc(1)) , (xc(2)-xc(1)) )
        angle   = theta*180/pie
      endif

200   continue

      return
      end
*********************************************************************
```

Appendix B. Input to "cplot". File "cplot.in".

```
temp
/users/luccio/plots/data/linear.pdat
/users/luccio/plots/data/kick_layout.pdat
/users/luccio/orbit/kick.pdat
/users/luccio/orbit/orbit.a.pdat
/users/luccio/orbit/orbit.w.pdat
/users/luccio/orbit/orbit.z.pdat
```

Appendix C. Parameter file containing the namelist $plotp. This file is transformed to "temp".

```
$plotp
 title='BNL AGS BOOSTER$'
 xx=9.
 yy=7.
 xlabel='s (m)$'
 ylabel='        x or y (mm)$'
 xpmin=-30.
 xpmax=240.
 ypmin=-15.
 ypmax=20.
 isym=3,
 n_data_max=1000
 max_plot=512
 n_y=2
 window=0,100,900,750
$
```

Appendix D. To plot a line graph. This makes a pair of curves for Fig. 1. The content of the first row, after "L" is ignored, as well as the informations after "Q".

```
L     s         xc        yc
      0.308     9.568    -1.004 MONV
      4.511    10.043    -0.105 MONH
      8.715    10.097     1.047 MONV
     12.919    10.459     0.835 MONH
     17.123    10.307     1.398 MONV
     21.326    10.133     0.226 MONH
     25.530     9.706    -1.039 MONV
     29.734     8.788    -1.027 MONH
     33.938     9.259    -2.101 MONV
     38.141     9.049    -0.808 MONH
     42.345     9.616    -0.149 MONV
     46.549    10.047     0.354 MONH
     50.753    10.247     1.179 MONV
     54.956    10.568     0.292 MONH
     59.160    10.504    -0.576 MONV
     63.364    10.980    -0.607 MONH
     67.568    10.414    -1.400 MONV
     71.771    10.552    -0.752 MONH
     75.975    13.940    -0.843 MONV
     80.179    21.633    -0.132 MONH
     84.383    14.783     0.240 MONV
     88.586    10.994     0.167 MONH
     92.790    10.254     0.357 MONV
     96.994     9.531     0.136 MONH
    101.198     9.435     0.068 MONV
    105.401     8.742    -0.079 MONH
    109.605     9.388    -0.475 MONV
    113.809     9.761    -0.560 MONH
    118.013    10.312    -1.183 MONV
    122.216    11.458    -0.550 MONH
    126.420    11.150    -0.469 MONV
    130.624    11.874    -0.064 MONH
    134.828    10.568     0.415 MONV
    139.031     9.783     0.076 MONH
    143.235     9.480    -0.557 MONV
    147.439     8.472    -0.542 MONH
    151.643     8.893    -0.909 MONV
    155.846     8.454    -0.298 MONH
    160.050     9.441     0.230 MONV
    164.254    10.076     0.662 MONH
    168.458    10.461     1.720 MONV
    172.661    11.410     1.079 MONH
    176.865    10.918     1.320 MONV
    181.069    10.837     0.125 MONH
    185.273     9.886    -1.223 MONV
    189.476     8.868    -1.254 MONH
    193.680     9.182    -2.470 MONV
    197.884     8.720    -1.087 MONH
Q   tunes              B_rho
 4.731051 4.692142 2.148156
 beta_x_max= 13.49146; beta_y_max= 13.29091
 xcmax=      11.63300; xcmin= -1.546000
 ycmax=      1.720000; ycmin= -2.470000
 <x>=  0.3966875   s(x)=  2.025079
 <y>=  -0.2153542   s(y)=  0.8719825
```

Appendix E (partial). To plot rectangles and segments. To plot the Booster layout of Fig. 1

```
P
  -4.00000000   0.0000000    35.00000   0.0000000
  -4.00000000   -10.00000    35.00000   0.0000000
   0.000000000  -15.00000    35.00000    90.00000
R
  -0.25187500 -12.00000  0.5037500   2.000000  0.0000000   QV
   3.95187500 -10.00000  0.5037500   2.000000  0.0000000   QH
   4.75562500 -11.50000   2.400000   3.000000  0.0000000   DH
   8.15562500 -12.00000  0.5037500   2.000000  0.0000000   QV
   8.95937500 -11.50000   2.400000   3.000000  0.0000000   DH
   12.3593750 -10.00000  0.5037500   2.000000  0.0000000   QF
   16.5631250 -12.00000  0.5037500   2.000000  0.0000000   QV
   17.3668750 -11.50000   2.400000   3.000000  0.0000000   DH
   20.7668750 -10.00000  0.5037500   2.000000  0.0000000   QH
   21.570625  -11.50000   2.400000   3.000000  0.0000000   DH
   24.9706250 -12.00000  0.5037500   2.000000  0.0000000   QV
   25.7743750 -11.50000   2.400000   3.000000  0.0000000   DH
   29.1743750 -10.00000  0.5037500   2.000000  0.0000000   QH
   29.9781250 -11.50000   2.400000   3.000000  0.0000000   DH
P
   33.6300000 -15.00000   35.00000   90.00000
R
   33.3781250 -12.00000  0.5037500   2.000000  0.0000000
   37.5818750 -10.00000  0.5037500   2.000000  0.0000000
   38.3856250 -11.50000   2.400000   3.000000  0.0000000
   41.7856250 -12.00000  0.5037500   2.000000  0.0000000
   42.5893750 -11.50000   2.400000   3.000000  0.0000000
   45.9893750 -10.00000  0.5037500   2.000000  0.0000000
   50.1931250 -12.00000  0.5037500   2.000000  0.0000000
   50.9968750 -11.50000   2.400000   3.000000  0.0000000
   54.3968750 -10.00000  0.5037500   2.000000  0.0000000
   55.2006250 -11.50000   2.400000   3.000000  0.0000000
   58.6006250 -12.00000  0.5037500   2.000000  0.0000000
   59.4043750 -11.50000   2.400000   3.000000  0.0000000
   62.8043750 -10.00000  0.5037500   2.000000  0.0000000
   63.6081250 -11.50000   2.400000   3.000000  0.0000000
P
   67.2600000  -15.00000   35.00000   90.00000
R
   67.0081250 -12.00000  0.5037500   2.000000  0.0000000
   71.2118750 -10.00000  0.5037500   2.000000  0.0000000
   72.0156250 -11.50000   2.400000   3.000000  0.0000000
   75.4156250 -12.00000  0.5037500   2.000000  0.0000000
   76.2193750 -11.50000   2.400000   3.000000  0.0000000
   79.6193750 -10.00000  0.5037500   2.000000  0.0000000
   83.8231250 -12.00000  0.5037500   2.000000  0.0000000
   84.6268750 -11.50000   2.400000   3.000000  0.0000000
   88.0268750 -10.00000  0.5037500   2.000000  0.0000000
   88.8306250 -11.50000   2.400000   3.000000  0.0000000
   92.2306250 -12.00000  0.5037500   2.000000  0.0000000
   93.0343750 -11.50000   2.400000   3.000000  0.0000000
   96.4343750 -10.00000  0.5037500   2.000000  0.0000000
   97.2381250 -11.50000   2.400000   3.000000  0.0000000
P
   100.890000  -15.00000   35.00000   90.00000
R
   100.638125 -12.00000  0.5037500   2.000000  0
   104.841875 -10.00000  0.5037500   2.000000
   105.645625 -11.50000   2.400000   3.000
   109.045625 -12.00000  0.5037500
   109.849375 -11.50000   2.400000
   113.249375 -10.00000  0.503
   117.453125 -12.00000  0
   118.256875 -11.500
   121.656875 -10
   122.460625
   125.860
   126
```

Appendix F. To plot rectangles and circles. This is used for the "circular layout" of Fig. 2. The coordinates are obtained by trigonometry from the "linear layout".

```
R
 -0.2461051   31.61304  0.5000000  0.7500000   0.4460343   1 qd
  0.5414103   31.60936   2.400000   1.000000  -0.9812753   2 b
   3.909375   31.62327  0.5000000  0.7500000   -7.047342   3 qf
   4.659013   31.26881   2.400000   1.000000   -8.474651   4 b
   7.937262   30.60138  0.5000000  0.7500000   -14.54072   5 qd
   11.95404   29.53668  0.5000000  0.7500000   -22.03409   6 qf
   12.58652   29.00042   2.400000   1.000000   -23.46140   7 b
   15.58067   27.50796  0.5000000  0.7500000   -29.52746   8 qd
   16.26102   27.11133   2.400000   1.000000   -30.95478   9 b
   19.18549   25.44074  0.5000000  0.7500000   -37.02084  10 qf
   22.16414   22.54320  0.5000000  0.7500000   -44.51422  11 qd
   22.71878   21.98412   2.400000   1.000000   -45.94152  12 b
   25.11177   19.61411  0.5000000  0.7500000   -52.00759  13 qf
   25.39175   18.83358   2.400000   1.000000   -53.43490  14 b
   27.25479   16.01940  0.5000000  0.7500000   -59.55450  15 qd--------------
   27.64536   15.33555   2.400000   1.000000   -60.98180  16 b
   29.34136   12.42574  0.5000000  0.7500000   -67.04787  17 qf
   29.40920   11.59931   2.400000   1.000000   -68.47518  18 b
   30.47028   8.426539  0.5000000  0.7500000   -74.54124  19 qd
   31.55657   4.415535  0.5000000  0.7500000   -82.03463  20 qf
   31.40839   3.599674   2.400000   1.000000   -83.46193  21 b
   31.61292  0.2604267  0.5000000  0.7500000   -89.52800  22 qd
   31.60960 -0.5270912   2.400000   1.000000   -90.95531  23 b
   31.62504  -3.895047  0.5000000  0.7500000   -97.02137  24 qf
   30.60498  -7.923398  0.5000000  0.7500000   -104.5147  25 qd
   30.39812  -8.683267   2.400000   1.000000   -105.9421  26 b
   29.54209  -11.94065  0.5000000  0.7500000   -112.0081  27 qf
   29.00612  -12.57338   2.400000   1.000000   -113.4354  28 b
   27.50045  -15.59390  0.5000000  0.7500000   -119.5550  29 qd--------------
   27.10350  -16.27406   2.400000   1.000000   -120.9823  30 b
   25.43151  -19.19772  0.5000000  0.7500000   -127.0484  31 qf
   24.74971  -19.66969   2.400000   1.000000   -128.4757  32 b
   22.53253  -22.17498  0.5000000  0.7500000   -134.5418  33 ~
   19.60202  -25.12121  0.5000000  0.7500000   -142.0351
   18.82136  -25.40081   2.400000   1.000000   -143                    ----------
   16.03175  -27.24753  0.5000000  0.7500000                        line b #1
   15.34807  -27.63841   2.400000   1.000                          q  2
   12.43903  -29.33573  0.5000000                                  q  3
   8.440332  -30.46647  0.5000                          .090       q  4
   7.678844  -30.66725                                 92.11690    q  5
   4.429836  -31.55                                   -52.11690    q  6
   3.613896                                  .00000    -52.11690    b #7
  0.245                                     1.000000   -36.34690    b #8
                                  .0000     1.000000   -4.806900    q 9
                                .5000000    1.000000    10.96310    b #10
                      .0      1.143000     1.000000    26.73310    q 11
                  .42496      0.5000000    1.000000    42.50310    b #12
              -12.48436       1.143000     1.000000    58.27310    q 13
     .08089   -9.201262       0.5000000    1.000000    74.04310    q 14
   -34.23007  -7.951208       0.5000000    1.000000    74.04310    q 15
   -33.59777  -5.739834       0.5000000    1.000000    74.04310    q 16
   -33.24038  -4.489927       0.5000000    1.000000    74.04310    q 17
   -32.60808  -2.278578       0.5000000    1.000000    74.04310    q 18
   -32.03076  -0.259497       0.5000000    1.000000    74.04310
C
   32.11400   0.0000000   32.11400   32.11400   0.0000000   360.0000
P
  0.0000000    35.32540   70.65079  -90.00000
  -30.59269   -17.66270   70.65079   30.00000
  -30.59269    17.66270   70.65079  -30.00000
  -56.19433    2.628707   40.00000  -59.61690   inj. line
  -51.13654   -5.997922   12.02995  -52.11690              1 to  7
  -43.74952   -15.49274   1.633728  -20.57690              7 to  8
  -42.22002   -16.06694   2.234032   10.96310              8 to 10
  -40.02676   -15.64207   5.656758   42.50310             10 to 12
  -35.85637   -11.82020   14.00602   74.04310             12 to 19
Q
//node_140c1/users/luccio/plots/data
Thursday, January 5, 1989   11:17:26 am (EDT)
```

Appendix G (partial). To plot text. For Fig. 1

```
P
  -4.00000000  0.0000000   35.00000  0.0000000
  -4.00000000  10.000000   35.00000  0.0000000
T
  -28.0000000   11.0000  0.5000000  0.0000000
HORIZONTAL$
  -28.0000000    1.0000  0.5000000  0.0000000
VERTICAL$
   15.0000000   -6.0000  0.5000000  0.0000000
A$
   4.00200000   10.0000  0.4000000  0.0000000   HK1A
1$
   11.4090000   10.0000  0.4000000  0.0000000   HK2A
2$
   20.8170000   10.0000  0.4000000  0.0000000   HK3A
3$
   29.2240000   10.0000  0.4000000  0.0000000   HK4A
4$
   49.0000000   -6.0000  0.5000000  0.0000000
B$
   37.4280000   10.0000  0.4000000  0.0000000   HK1B
1$
   45.8360000   10.0000  0.4000000  0.0000000   HK2B
2$
   54.2430000   10.0000  0.4000000  0.0000000   HK3B
3$
   61.6510000   10.0000  0.4000000  0.0000000   HK4B
4$
   83.0000000   -6.0000  0.5000000  0.0000000
C$
   70.8540000   10.0000  0.4000000  0.0000000   HK1C
1$
   79.2620000   10.0000  0.4000000  0.0000000   HK2C
2$
   87.6690000   10.0000  0.4000000  0.0000000   HK3C
3$
   96.0770000   10.0000  0.4000000  0.0000000   HK4C
4$
   115.000000   -6.0000  0.5000000  0.0000000
D$
   104.281000   10.0000  0.4000000  0.0000000   HK1D
1$
   111.688000   10.0000  0.4000000  0.0000000   HK2D
2$
   120.596000   10.0000  0.4000000  0.0000000   HK3D
3$
   129.503000   10.0000  0.4000000  0.0000000   HK4D
4$
   150.000000   -6.0000  0.5000000  0.0000000
E$
   137.707000   10.0000  0.4000000  0.0000000   HK1E
1$
   146.114000   10.0000  0.4000000  0.0000000   HK2E
2$
   154.522000   10.0000  0.4000000  0.0000000   HK3E
3$
   161.929000   10.0000  0.4000000  0.0000000   HK4E
4$
   182.000000   -6.0000  0.5000000  0.0000000
F$
   171.133000   10.0000  0.4000000  0.00
1$
   179.541000   10.0000  0.400
2$
   187.948000   10.00
3$
   196.35
4$
```

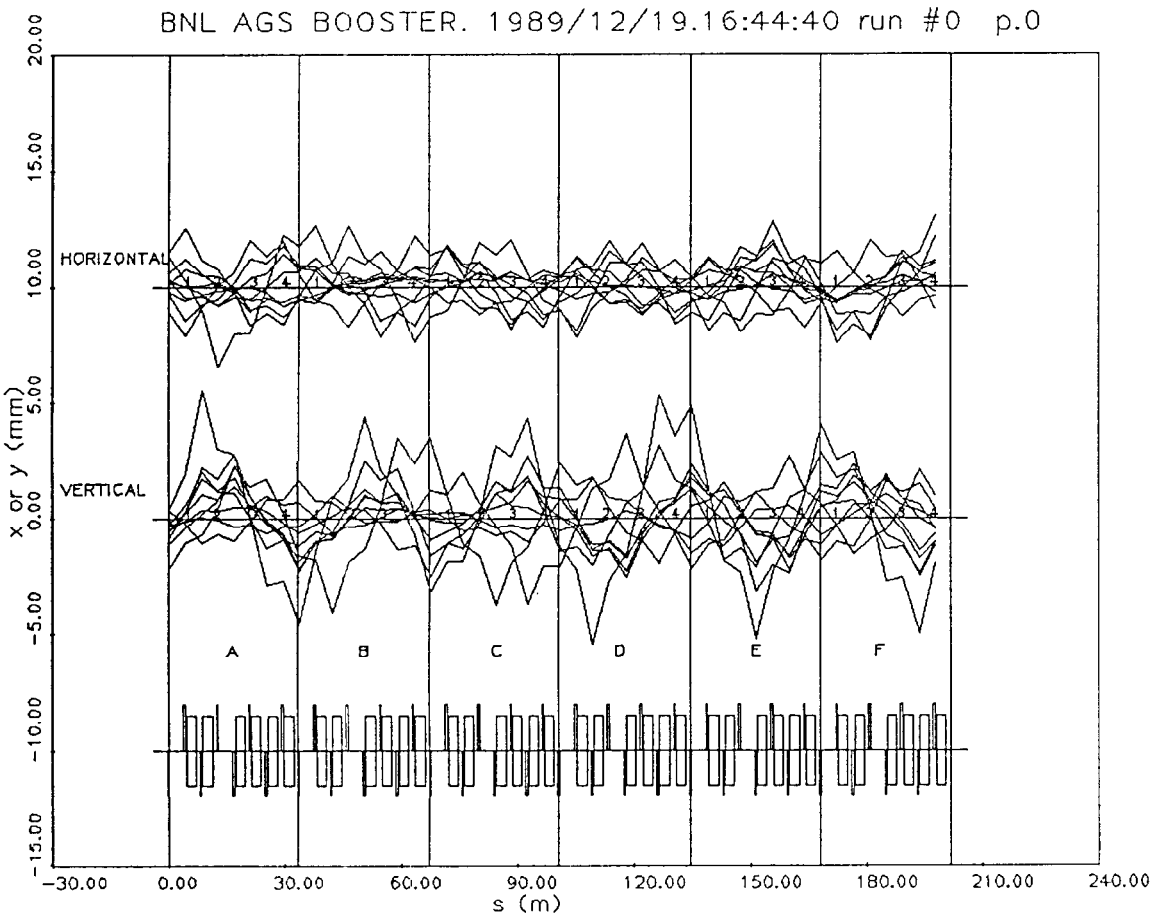

Fig. 1. A collection of distorted orbits in the Booster obtained by random extraction of errors of position, angle and field in the magnets.

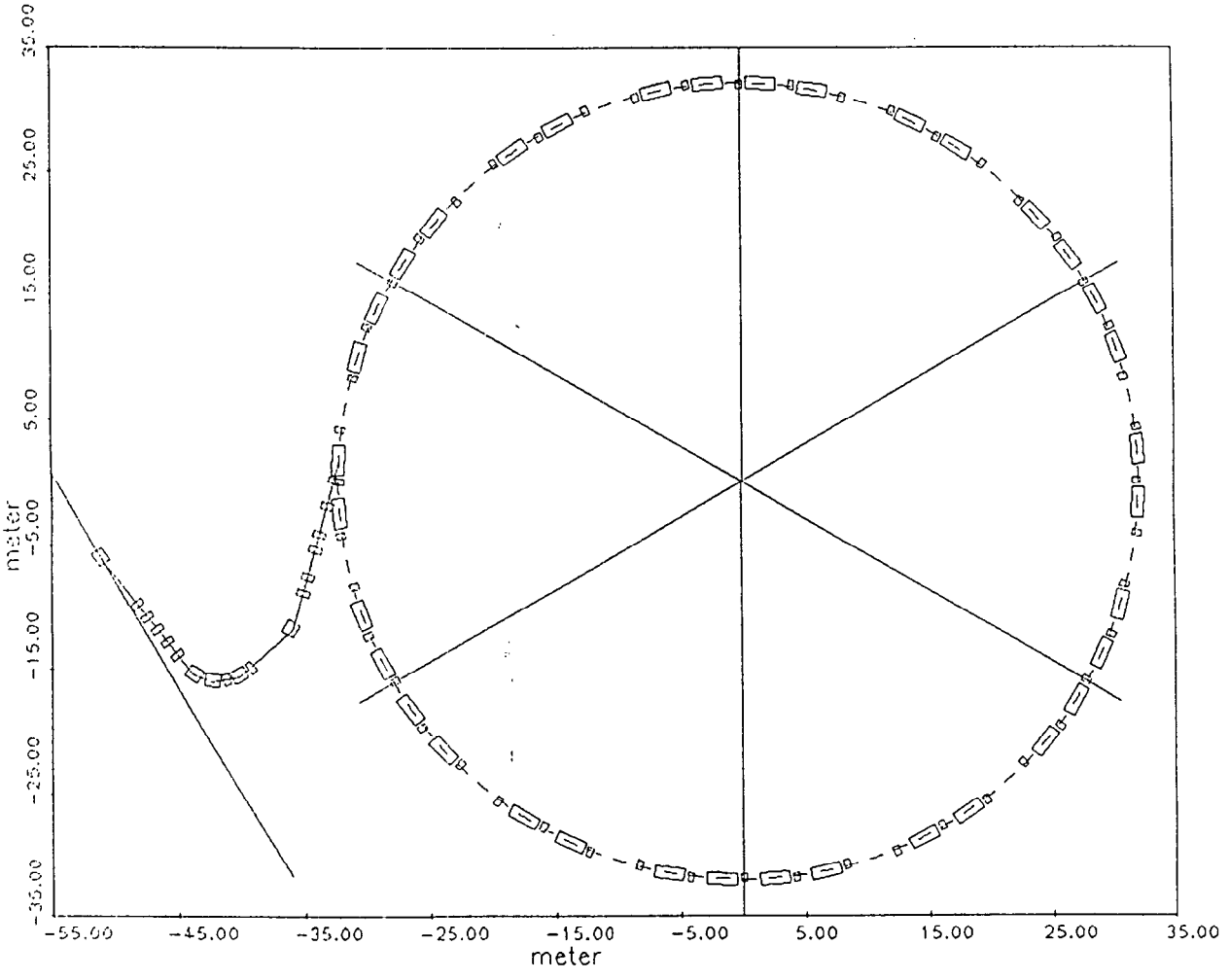

Fig. 2. "Circular layout" of the Booster and the proton injection line.

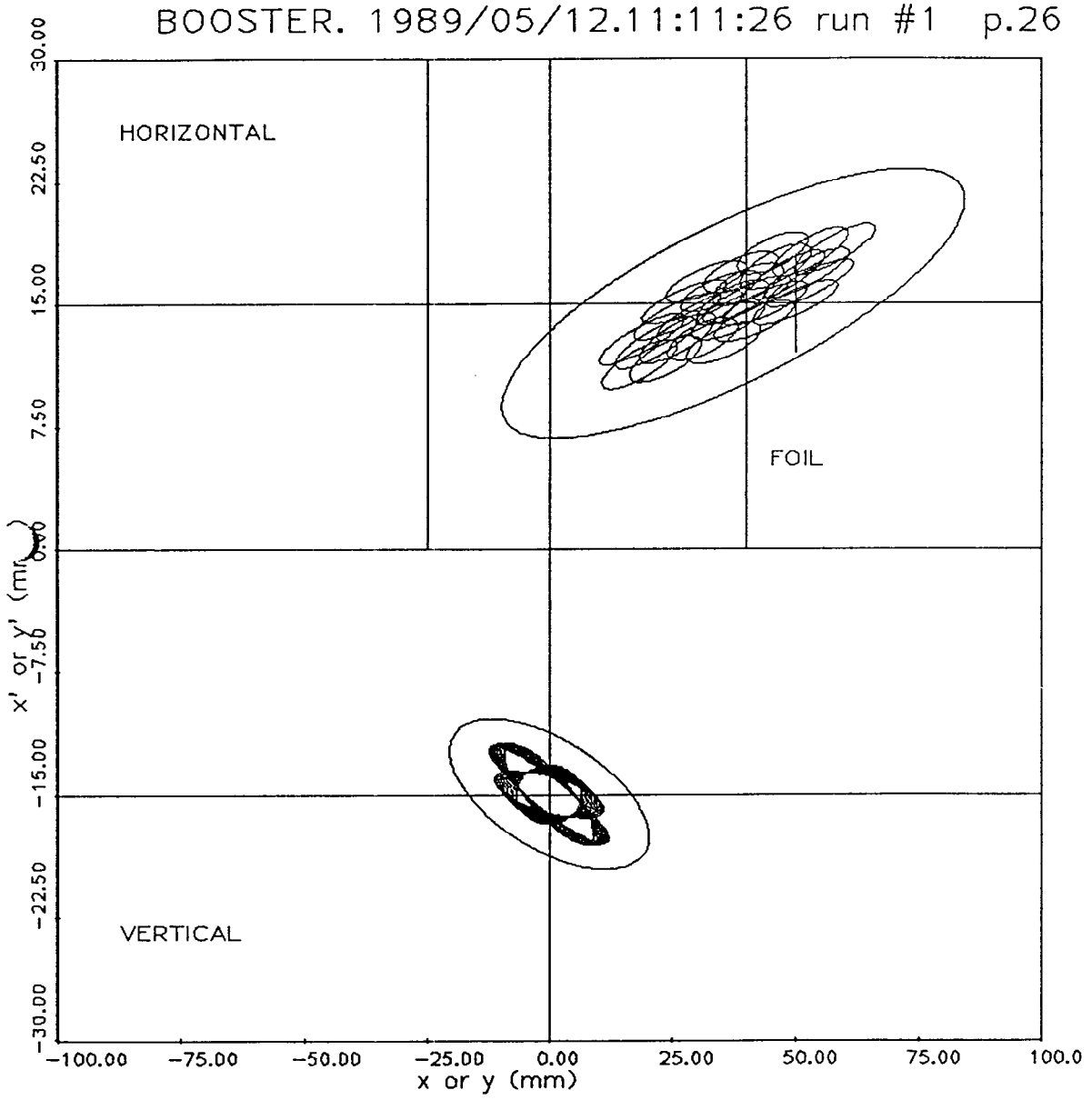

Fig. 3. Injection of protons in the Booster. The acceptance phase space ellipses and the injected proton emittances are shown.