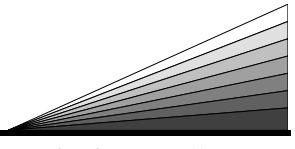
Xorbit Reference Manual

Kenneth Evans, Jr.

July 1994

ARGONNE NATIONAL LABORATORY

9700 South Cass Avenue, Argonne Illinois 60439



Advanced Photon Source

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ABSTRACT

Xorbit is an accelerator physics code that tracks particle orbits. Its two distinguishing features are a rich graphical interface and the ability to connect to and be controlled by external programs such as Mathematica or control-system software. The design goal is to have a code that can be controlled in much the same way as a real machine is controlled. This allows the testing of control algorithms before the real machine is commissioned or without disturbing the real machine at any time. The graphical interface provides a means of changing magnet parameters easily with immediate visual feedback on the resulting orbit changes. There are a number of features including interactive plotting of orbits and Twiss parameters; the ability to display error positions, monitor readings, or the full orbit; the ability to display true or difference orbits; as well as the ability to find closed orbits, track from given initial conditions, or apply a variety of correction methods. There is a design mode in which element strengths and positions may be changed with the mouse with continuous display of the results. All of these operations are fast and intuitive.

TABLE OF CONTENTS

1. Introduction 1
- 1.1 Overview 1
- 1.2 Starting Xorbit 1
- 1.3 The Graphic Interface 2
- 1.4 The Icon Area 3
  - Mouse Operations 3
  - Controls 4
- 1.5 The Graph Area 4
  - Mouse Operations 4
  - Controls 4
- 1.6 The Plot Window 5
  - Mouse Operations 6
  - Controls 6

2. The Input File 7
- 2.1 Title Line 7
- 2.2 Input NAMELIST 7
- 2.3 Element Specification 7
  - Injection Point 8
  - Monitor 8
  - Label or Marker 8
  - Drift Length or Straight 8
  - Bend 9
  - Edge 9
  - Nonlinear or Multipole 9
  - Quadrupole or Combined Function 10
  - Thin 10
- 2.4 Option Blocks 10
- 2.5 Ending Line 10
- 2.6 Sample Input File 11
- 2.7 Converting from Mad 13
  - Directions 13

3. Main Menu Bar 15
- 3.1 File Menu 15
- 3.2 Options Menu 15
- 3.3 Orbit Menu 16
- 3.4 Clear Menu 17
- 3.5 Help Menu 17

4. Plot Menu Bar 19
- 4.1 Options Menu 19
- 4.2 Clear Menu 19
- 4.3 Help Menu 19

5. Dialog Boxes 20
- 5.1 Orbit options 20
  - Top Controls 20
  - Plot: 20
  - BPM Only: 20
  - Start Orbit From: 21
  - x0:, x0':, y0:, and y0': 21
  - Orbit Turns: 21
  - Xvgr Plot: 21

First Turn: 21
5.2 Plot Options 22
Top Controls 22
Options: 22
Table Controls 22
5.3 Design Mode 22
Top Controls 23
Options: 23
Attach To: 23
Position Change: 23
Value Change: 24
Current Value: 24
Current Element: 24
Arrow Keys 24
5.4 Element Information 24
Top Controls 24
5.5 Initial Twiss 24
Top Controls 25
6. Input Switches 26
6.1 General 26
6.2 Channel Access 26
6.3 Correctors 27
6.4 Orbit 27
6.5 Plots 28
7. Option Blocks 29
7.1 Grouping of Elements 29
7.2 First-Turn Correction 29
7.3 Quadrupole-Movement Correction 30
7.4 Girder-Movement Correction 30
8. Channel Access 31
8.1 How Channel Access is Emulated 31
8.2 Mathematica Interface to Channel Access 32
Directions 32
CaGet and CaSet 33
8.3 Technical Details 33
9. Description Of The Code 36
Files 36

1. Introduction

1.1 Overview

Xorbit is a computer code designed to help in the design and operation of circular accelerators, storage rings, and beam lines. It has been written for Unix systems using a combination of Fortran and C. The physics code is mostly Fortran and the interface is mostly C. The code has been developed on a Sun workstation and should work well in that environment The code has been used with the Motif window manager, Mwm, with the VUE window manager, Vuewm, with the X Windows window manager, Twm, as well as with the Open Windows window manager, Olwm. If Sun's Open Windows is used, it must be Open Windows 3 or newer, since there were bugs in the server for Open Windows 2 that did not allow Motif applications in general to work correctly.

If these terms are not familiar to you, you should talk to your system administrator, read one of the general-information books about X Windows, or otherwise develop at least a passing acquaintance with X Windows on *your* computer. If you do not use the graphical interface, there is probably little to recommend Xorbit over the more established accelerator codes. Xorbit is an accelerator code for the '90's.

Xorbit may be operated as a program that reads an input file and produces several output files without further user intervention, or it may be operated with an X-Windows, interactive interface allowing continuous and varied user input with a display of the orbit and other information as they the information is generated. It has a **Channel Access** capability that allows it to truly emulate a real device and to be controlled in much the same manner as a real device. This capability is useful for testing control algorithms and other typical accelerator operations. The interactive interface provides a visualization of the orbit that is not typically available in other accelerator codes.

1.2 Starting Xorbit

In order to run Xorbit the application resource file, EvansApp, must be in one of the standard places for resource files. Put it in your home directory if you are not sure or if you expect to modify it. The executable file, xorbit, must be in your path, and Xorbit must know where to find the help files. You must have an input file.

HyperHelp should be set up on your system, and the file, HyperHelp, which contains additional resources for HyperHelp, should be in your home directory. By default Xorbit will look for xorbit.hlp in the current working directory. It will look for the HyperHelp viewer, hyperhelp, in /usr/bin/X11, and it will look for the help-on-help file, hoh.hlp, in /usr/lib/hyperhelp. These three default locations can be changed by setting the environment variables, XORBITHOME, HHPATH, and HOHPATH, respectively. If HyperHelp is not set up or the help files are not in the right places, Xorbit will still work but will have a only a minimal help package.

In addition, programs such as XTerm, Mathematica, and Xvgr that may be available from the Xorbit menus must be in your path if you wish to use them.

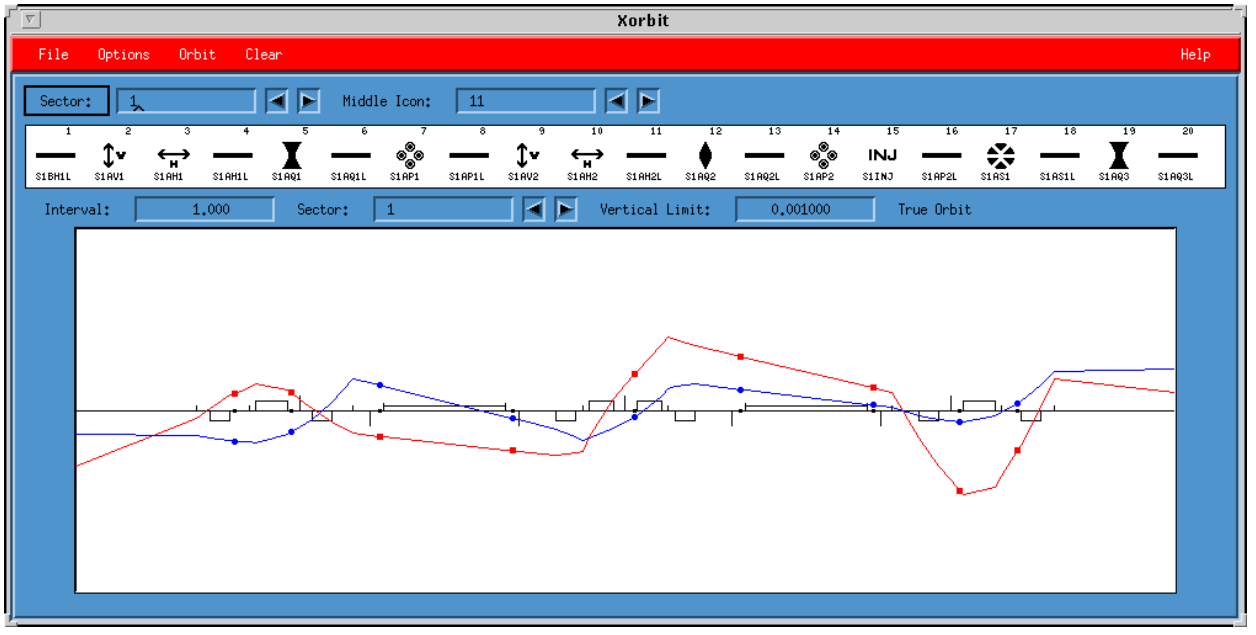
The code is executed by typing:

xorbit [<inputfile>] [&]

The []'s indicate optional parameters. The <inputfile> parameter is the first part of the name of an existing input file whose full name is <inputfile>.in, *e.g*. aps.in. If <inputfile> is omitted, then default.in is used if it exists. A switch. IINTERACTIVE, in the input file determines whether the interactive interface is displayed or not. The & is used in Unix to put the job in the background, a good thing to do when Xorbit is run interactively.

An output file with the name <inputfile>.out will always be produced, and other output, e.g. plot, files whose names start with <inputfile> may also be produced.

The first time Xorbit is used, it would probably be a good idea to use an input file that is known to work and to just explore the interface. Most of the operations should be intuitive; they were designed with that in mind.



Xorbit Main Window

1.3 The Graphic Interface

Most of the work in Xorbit is done in its Main Window. The Menu Bar is at the top of the main window, with buttons labeled File, Options, Orbit, Clear, and Help. Below the Menu Bar is the Icon Control Area. This area contains the controls for manipulating the Icon Area, which is just below it. The icon area displays icons representing the magnet elements. Below the icon area is the Graph Control Area. This area contains controls for adjusting and scrolling the Graph Area, the bottom area, which displays the orbits.

In order to use the graphic interface of Xorbit, it is of course necessary to have an X window manager running before Xorbit is started. The frame around Xorbit windows will vary depending on the window manager used. The frame shown is that supplied by Olwm, the Open Look window manager, which typically comes with Sun workstations. The Xorbit interface, however, is a Motif interface. Any X application, whether it be a Motif application, an Open Windows application, or another X application, *should* run under any X window manager. As X and its applications become more robust, this coexistence is becoming more of a reality. Some of the things you can do with Xorbit, nevertheless, are different with different window managers.

It is assumed that you are familiar with Graphical User Interfaces (GUI) such as Motif, Open Windows, Microsoft Windows, or the Macintosh Operating System. It would help to be familiar with Motif, in particular, but the idea is that if you know one GUI, then you should be able to use another with a minimum of difficulty. We will describe a few of the Motif conventions here, hopefully enough to be able to use Xorbit effectively.

Motif uses a three-button mouse. The buttons may be referred to as:

Select	Left Button	Button 1
Drag	Middle Button	Button 2
Custom	Right Button	Button 3

In this document the buttons will be referred to as **Button 1**, **Button 2** and **Button 3** which are normally the left, middle, and right buttons, respectively. If you have redefined the buttons on your mouse, you will have to adjust accordingly. **Click** means quickly press and release the button. **Double Click** means click the button twice in rapid succession without moving the mouse. **Drag** means press the button and move the mouse while holding the button down. Button 1 is used for most operations in the Motif interface. If no button is specified use Button 1. In particular Button 1 is used to select items. Pull-down menus, such as those on the Menu Bar may be displayed by clicking or dragging down on their push buttons. If you clicked the button, you make the menu choice by clicking on the desired menu item. If you dragged, you release the mouse button over the item you want to select. You cancel the operation without making a choice by dragging or clicking somewhere off the menu you have displayed. Menus may have items that perform an action, items that can be dragged right to display sub-menus, or items that display a **Dialog Box** that allows further operations. Drag-right menus are indicated by an arrow, and dialog boxes are indicated by an ellipsis (...).

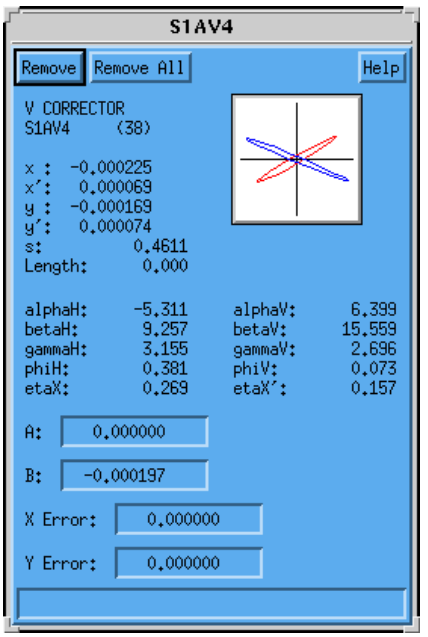
List Boxes, such as in the icon and graph control areas, are places where you can enter text. The existing text can be **Selected** by dragging over it or double clicking on it. Text that is selected will be replaced by whatever is typed on the keyboard. Text can also be inserted or deleted by positioning the mouse pointer in the desired location and typing or using the <Back Space> key, or possibly the <Delete> key. In Xorbit the text is not entered until the <Return> key is pressed, so be careful that your changes are really entered and not just left showing in the list box.

The basic user operations with the Xorbit interface include clicking and dragging on items with the mouse, entering and editing text in the list boxes, and clicking and dragging the mouse in the icon and graph areas.

> The best way to get familiar with the interface is to use it! You can pick it up much faster that way than by reading about it. One of the main advantages of any GUI is that you can poke around it, pulling down menus and opening dialog boxes to see what is there. With a little common sense it is unlikely you will permanently damage anything. You can always refer to the manual or the on-line help to clear up details that are not obvious from poking around.

1.4 The Icon Area

The icon area contains icons for each of the elements in the lattice. Both the index of the element in the lattice and its name are shown. The display can be scrolled in three main ways: The sector may be changed by entering a new sector in the list box labeled Sector or by clicking on the arrows after this list box. These arrows scroll the display by one sector at a time while staying at the same relative position in the sector. If there is an ambiguity *sector* refers to the sector of the icon at the middle. The icon which is at the middle may also be changed by entering its lattice number in the list box after Middle or by clicking the arrows after this list box. Clicking these arrows scrolls the display by one icon at a time. Finally, if an element is clicked with Button 1 in the graph area, the icon for that element will appear at the middle of the icon area display. This is the easiest way to pick icons.



Element Information Dialog Box

Mouse Operations

The main reason for the icon area is that if you click on an icon in the icon area, a dialog box will pop up giving information about that element. This information is current at the time the icon is clicked and will not be updated. Clicking at a later time on the icon for the same element will display another box with the then-current information. A fairly large number of boxes may be displayed at once. They may be

dragged around the screen and placed as desired. They may be used as a means of keeping track of parameters as changes are made in the orbit. For more information on the contents of this dialog box see the section on dialog boxes.

Controls

Sector:

The icon bar will move to the sector entered in the text box (after pressing <CR>). The arrows by the text box will scroll the icon bar by a sector.

Middle Icon:

The icon bar will move so that the icon for the element with the number entered in the text box (after <CR>) is in the center of the bar. That icon will be highlighted. Also the arrows by the text box will scroll the icon bar by one element and highlight the middle icon.

1.5 The Graph Area

The graph area contains the orbit display. The horizontal axis is s, the longitudinal distance along the lattice, and the vertical axis is x or y, the transverse coordinates. Exactly what is displayed may be controlled via the Orbit Options item in the Options pull-down menu, and the area may be cleared via items in the Clear pull-down menu. These menu items are described in detail in their own sections. In addition to the menu controls, there are controls in the graph control area and button operations in the graph area itself.

Mouse Operations

There are three operations that may be performed with the mouse in the graph area.

1. Clicking Button 1 moves the icon for the element at the mouse pointer to the middle of the icon area, where it may be clicked to display information about the element.
2. Clicking Button 2 moves the position which is at the mouse pointer to the center of the graph area. This is the means of adjusting what appears in the display interval. The graph area display may be scrolled discontinuously by repeatedly clicking Button 2. The closer you are to the center, the more slowly the display scrolls. The direction depends on which side of center you are.
3. Pressing and holding Button 3 scrolls the lattice continuously. The speed depends on how far you are from the center, and the direction depends on which side of center. The mouse may be moved while the button is down to vary the speed and direction.

Controls

Interval:

The number in the text box determines the length of the interval displayed in the graph area. (See ISUNITS in the input switches.) The interval will not change until a <CR> is pressed. The interval may be longer or shorter than a sector.

Sector:

The graph area will move so that the sector entered in the text box is in the center of the graph area (after <CR>). Also the arrows by the text box will scroll the graph area by an interval.

Vertical Limit:

The number in the text box represents the vertical limits (in meters) of the orbit plot in the graph area. The vertical axis extends from minus this number to plus this number.

Reference Orbit

The text to the right of the Vertical Limit control identifies if the plots are a difference orbit or not. If not, it says "True Orbit". Otherwise is says "Difference w.r.t. n", where n is the number of the difference orbit. See the orbit menu section for more information on difference orbits.

The starting value for the display interval is one sector and all of Sector 1 is displayed. With these values you can scroll through the lattice, seeing a full sector at a time. The sector midpoint is always at the center of the display. This is probably what you usually want to do. If Buttons 2 or 3 are used in the graph area, the center of the sector will typically not continue to be at the center of the display. Also if the interval is set to a fraction of a sector and then returned to a full sector, the center of the sector may not be at the center of the display. To reset the center of the sector to the center of the display, click with Button 2 on the center of the sector.

The x and y coordinates are shown in different colors, red for x and blue for y by default on a color monitor and solid and dashed on a monochrome monitor. When the Xorbit window is resized, the change is predominately in the graph area.

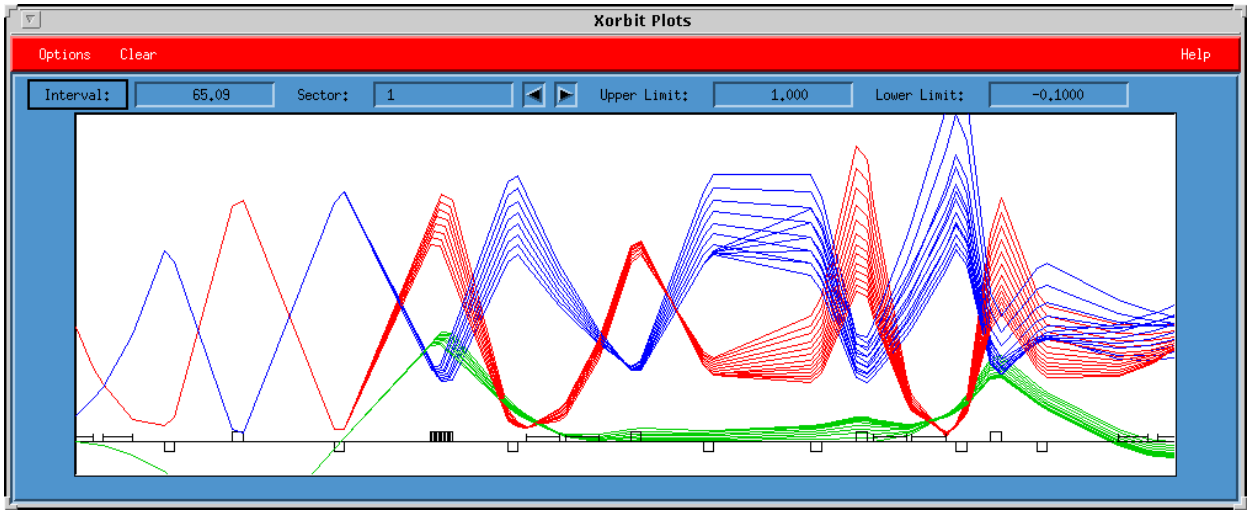
1.6 The Plot Window

Plots of a number of parameters vs. s may be displayed in the plot graph area in the plot window. What is displayed and how it is displayed are controlled by switches in the input file and by the plot options dialog box. These options are described in more detail in the section on dialog boxes.

The parameters that may be displayed as a function of s are:

α_h	α_v
β_h	β_v
γ_h	γ_v
η_x	η'_x
x	y
x'	y'
x_{err}	y_{err}

There is a design mode that allows the element strengths and positions to be changed by clicking with the mouse or by entering values in a text box. The plots in the plot window change as the elements are changed, allowing immediate visual feedback of the results of the changes on any of the above parameters. The operation of this mode is described in detail in the section on dialog boxes.



Xorbit Plot Window

Mouse Operations

The mouse operations are the same as those for the graph area in the main window: Clicking Button 1 on a element (actually on the value of s for that element) will cause the icon display to move to that icon. Clicking Button 2 on an element will cause that element to move to the center of the display. Holding down Button 3 will cause the display to scroll. The scrolling speed depends on the distance of the mouse cursor from the midpoint.

Controls

Interval:

The number in the text box determines the length of the interval displayed in the graph area. (See ISUNITS in the input switches.) The interval will not change until a <CR> is pressed. The interval may be longer or shorter than a sector.

Sector:

The graph area will move so that the sector entered in the text box is in the center of the graph area (after pressing <CR>). The arrows by the text box will scroll the graph area by an interval.

Upper Limit:

The number in the text box represents the upper limit (in meters) of the plots in the graph area. The vertical axis extends from the lower limit (see below) to this number.

Lower Limit:

The number in the text box represents the lower limit (in meters) of the plots in the graph area. The vertical axis extends from this number to the upper limit (see above).

2. The Input File

The input file must have a name that ends in .in. The first part of the name (excluding the .in) may be given as the first argument on the command line. Otherwise, default.in is used. if it exists.

The structure of the input file is as follows:

One Title line (Any text)

Input NAMELIST (See below and the section on switches, End with " $")

Two alignment guide lines (For entering the elements)

List of elements and their parameters (See below, End with " $")

Zero or more Option Blocks (Start with "**name*", End with " $")

Ending line ("*End")

There is a sample input file below in this section.

2.1 Title Line

The first line of the input file is a title line. It may be anything you like up to 80 characters. Any more will be ignored. This title appears in the message Xorbit writes on the screen when it starts, in the output file, and as the subtitle on plots.

2.2 Input NAMELIST

The second block of input is the Xorbit NAMELIST. According to Sun conventions it must start with $INPUT in column 2 or beyond of the first line and end with $ or $END in column 2 or beyond of the last line. In between you may enter switches and their values in the form SWITCH=VALUE, e.g. iinteractive=1 with white space or newlines between them, again avoiding column 1. Case is not significant. Switches which are not entered will have their default values. You only *need* to specify a switch setting to change it or remind yourself what it is. See the sample input for an example. The possible switches are given in the section on Switches. The switch setting IECHOSW=1 will echo the NAMELIST (in an unattractive form) in the output file.

2.3 Element Specification

An element in Xorbit is defined by the quantities: name, type, a, b, slen, xbar, ybar, and ysig. For a multipole there is an additional quantity, npole, which is the multipole order. For a bend there are additional quantities e1, e2, fint, and hgap. In most cases a is the normal strength of the element and b is the skew strength, but these quantities are different for a bend and for bend edges. slen is the length of the element, xbar is the mean displacement in the x direction, and xsig is the standard deviation of the displacement. Similarly, for ybar and ysig. If xsig or ysig is non-zero, the corresponding displacement of that element is determined randomly from a Gaussian distribution. The input quantities are read in the following order with the following format:

name, type, npole, a, b, slen, xbar, xsig, ybar, ysig

FORMAT(A<namesize>,1X,A1,A1,1X,7F10.0)

The length of the names, namesize, is determined by the switch NAMESIZE.

For a bend there is an additional, continuation line for the additional bend parameters.

If a name is repeated in the list of elements, then the rest of its parameters are overwritten by the parameters for the first occurrence of that name. Consequently they do not need to be typed. Ignored, zero, or blank parameters also do not need to be typed.

The lines containing the element definitions are preceded by two lines which are ignored (and which can be used for alignment). The element definitions must be followed by a line containing " $" to terminate the element input. See the sample input file.

The possible elements and their inputs in detail are:

Injection Point

type: i

npole: ignored and set to blank

a: ignored and set to 0

b: ignored and set to 0

slen: ignored and set to 0

(There *must* be one injection point. It need not be at the start of the elements. Initial values for the orbits refer to this location in the lattice. For the first-turn algorithm, this point must be just before a monitor.)

Monitor

type: m

npole: ignored and set to blank

a: ignored and set to 0

b: ignored and set to 0

slen: ignored and set to 0

Label or Marker

type: l

npole: ignored and set to blank

a: ignored and set to 0

b: ignored and set to 0

slen: ignored and set to 0

Drift Length or Straight

type: s

npole: ignored and set to blank

a: ignored and set to 0

b: ignored and set to 0

slen: l

(l is the length of the drift space.)

Bend

type: b

npole: ignored and set to blank

a: θ_b / π

b: If non-zero, use this value as ρ and calculate l. Otherwise calculate ρ from slen.

slen: l if b is zero, otherwise calculated as $l = \rho\theta_b$.

(θ_b is the bend angle, ρ is the radius, and l is the pathlength. These are related by $l = \rho\theta_b$)

There is an optional second line for a bend. The second line is read with the following format:

continuation, e1, e2, fint, hgap

FORMAT(A<namesize>,4X,7F10.0)

The continuation string has an "&" in column 1 or 2. The quantities e1 and e2 are the edge angles, and fint and hgap are quantities that further define the kick caused by the magnet edge. The quantity hgap is equal to 0.5 g, where g is the width of the magnet gap, and fint is equal to K_1, which is defined as:

$$K_1 = \int_{-\infty}^{\infty} \frac{B_y(s)(B_0 - B_y(s))}{gB_0^2} ds$$

These quantities in Xorbit are the same as those defined in Mad, and a description and further references may be found in the Mad manual.

Edge

(The edge element is no longer available in Xorbit. All of the edge parameters are specified in the Bend element.)

Nonlinear or Multipole

type: n

npole: 1-9 (Dipole is 1)

a: A_n

b: B_n

slen: ignored and set to 0

(A_n and B_n are the coefficients in the expansion of the field:

$$B_x + iB_y = \frac{B_0\rho_0}{l}(-\Delta x' + i\Delta y') = \frac{B_0\rho_0}{l}\sum_n (A_n + iB_n)(x+iy)^{n-1} \ .$$

For $B_n = 0$, the kick is related to the field derivative,and K_nL as defined by Mad through:

$$\Delta x' = -A_n \,\mathrm{Re}(x+iy)^{n-1} = -\frac{l}{(n-1)!B_0\rho_0}\frac{\partial^{n-1}B_y}{\partial x^{n-1}}\mathrm{Re}(x+iy)^{n-1} = -\frac{1}{(n-1)!}K_{n-1}L\,\mathrm{Re}(x+iy)^{n-1}$$

$$\Delta y' = A_n \,\mathrm{Im}(x+iy)^{n-1} = \frac{l}{(n-1)!B_0\rho_0}\,\frac{\partial^{n-1}B_y}{\partial x^{n-1}}\mathrm{Im}(x+iy)^{n-1} = \frac{1}{(n-1)!}K_{n-1}L\,\mathrm{Im}(x+iy)^{n-1}$$

To tilt a multipole which has coefficients A_{n0} and B_{n0}=0 by an angle θ about the s axis, take A_n=A_{n0}cos(nθ) and B_n=-A_{n0}sin(nθ). Correctors are specified as n1 multipoles (dipoles). Note that for horizontal correctors A_n is (minus) the corrector strength in radians and that for vertical correctors B_n is the corrector strength in radians. Skew quadrupoles are specified as n2 multipoles with a B_n component.. For normal sextupoles (n3 multipoles), A_n is (B"l) / ($2B_0\rho_0$) = $K_2L/2$. (There are no finite-length, nonlinear elements in Xorbit.)

Quadrupole or Combined Function

type: q

npole: ignored and set to blank

a: K

b: ρ (Use 0 for a pure quadrupole with no bend)

slen: l

(K is the quadrupole strength, B' / ($B_0\rho_0$); ρ is the radius for a combined-function element; and l is the length.)

Thin

type: t

npole: ignored and set to blank

a: Kl

b: ignored and set to

slen: ignored and set to (K is the quadrupole strength and l is the length.)

2.4 Option Blocks

Xorbit is designed so that input for new options can be added easily in blocks. The start of each block is specified by a line containing "**name*" and ends with a line that contains " $". (The $ is in column 1 or 2.) The order of the option blocks is not important. Currently the possible *name*'s are:

group:	Grouping for elements with common power supplies
firstturn:	First-Turn Correction Algorithm
girdercorrected:	Girder Correction Algorithm
quadmovement:	Quad-Movement Correction Algorithm

(Case and spaces are ignored in *name*.) See the section on option blocks for the exact form of the input for each option and for more details. See the sample input for an example of an option block.

2.5 Ending Line

The last line must be:

*End

Case and spaces are ignored as for the option-block names. Any lines after this line are not read and may be used for any purposes you desire.

2.6 Sample Input File

The following is an input file for the APS Storage Ring. It has 40 identical sectors. The lines at the end are editing sequences for Emacs and are not read by Xorbit.

```
'93 APS Storage Ring Lattice
 $input
 iplcs=2
 iinteractive=1 icainit=1 icawait=0
 iseed=0
 nsect=40 isunits=1 namesize=4 nameprefix='S' inameconv=1
 xstart=.0001 xpstart=0. ystart=.0001  ypstart=0.
 idistcheck=1 ipelerr=1 nturns=10
 ifto=2 iplfto=0 iplevcor=1 iplevinj=1
 iclo=1 iplclo=0 iplevclo=1
 iplcorh=0 iplcorv=0 icorsort=3
 xwall=.08 ywall=.04
 itwissupdate=1 scalebetah=.04 scalebetav=.04
 scalealphah=.04 scalealphav=.04 scalegammah=.04 scalegammav=.04
 scalex=1000. scaley=1000. scalexp=1000. scaleyp=1000.
 scaleetax=.4 scaleetaxp=.4
 $
name tn      a         b       slen       xbar      xsig      ybar      ysig
a--- ai f---------f---------f---------f---------f---------f---------f---------
BH1L s                       3.0360
AV1  n1           0.0e-3
AH1  n1 0.0e-3
AH1L s                       0.3240
AQ1  q  -.45435995           0.50                5.0e-4              5.0e-4
AQ1L s                       0.1142
AP1  m                                           2.0e-4              2.0e-4
AP1L s                       0.3758
AV2  n1           0.0e-3
AH2  n1 0.0e-3
AH2L s                       0.1700
AQ2  q  .639333739           0.80                5.0e-4              5.0e-4
AQ2L s                       0.0847
AP2  m                                           2.0e-4              2.0e-4
INJ  i
AP2L s                       0.2153
AS1  n3 0.8286
AS1L s                       0.30
AQ3  q  -.41158941           0.50                5.0e-4              5.0e-4
AQ3L s                       0.5200
AV3  n1           0.0e-3
AH3  n1 0.0e-3
AH3L s                       0.4400
AS2  n3 -1.66585
AS2L s                       0.2310
AP3  m                                           2.0e-4              2.0e-4
AP3L s                       0.1190
AM   b  .025      38.961112
 &      .0125     .0125
AML  s                       0.1537
```

```
AP4  m                                          2.0e-4              2.0e-4
AP4L s                      0.1963
AS3  n3 -2.08915
AS3L s                      0.9100
AQ4  q  -.80954550          0.50                5.0e-4              5.0e-4
AQ4L s                      0.1750
AV4  n1           0.0e-3
AH4  n1 0.0e-3
AH4L s                      0.1750
AQ5  q  .780136057          0.60                5.0e-4              5.0e-4
AQ5L s                      0.3000
AS4  n3 1.9126
AS4L s                      0.2271
BP5  m                                          2.0e-4              2.0e-4
BP5L s                      0.0729
BQ5  q  .780136057          0.60                5.0e-4              5.0e-4
BQ5L s                      0.1750
BV4  n1           0.0e-3
BH4  n1 0.0e-3
BH4L s                      0.1750
BQ4  q  -.80954550          0.50                5.0e-4              5.0e-4
BQ4L s                      0.9100
BS3  n3 -2.08915
BS3L s                      0.2310
BP4  m                                          2.0e-4              2.0e-4
BP4L s                      0.1190
BM   b  .025      38.961112
 &      .0125     .0125
BML  s                      0.1536
BP3  m                                          2.0e-4              2.0e-4
BP3L s                      0.1964
BS2  n3 -1.66585
BS2L s                      0.2640
BV3  n1           0.0e-3
BH3  n1 0.0e-3
BH3L s                      0.6960
BQ3  q  -.41158941          0.50                5.0e-4              5.0e-4
BQ3L s                      0.3000
BS1  n3 0.8286
BS1L s                      0.2207
BP2  m                                          2.0e-4              2.0e-4
BP2L s                      0.0793
BQ2  q  .639333739          0.80                5.0e-4              5.0e-4
BQ2L s                      0.1700
BV2  n1           0.0e-3
BH2  n1 0.0e-3
BH2L s                      0.4008
BP1  m                                          2.0e-4              2.0e-4
BP1L s                      0.0892
BQ1  q  -.45435995          0.50                5.0e-4              5.0e-4
BQ1L s                       .3240
BV1  n1           0.0e-3
BH1  n1 0.0e-3
BH1l s                      3.0360
 $
*First turn
name t  strength  bpm1 bpm2 bpm3 bpm4 bpm5 bpm6 bpm7 bpm8 bpm9 bpm10 ...
a--- a f--------- a--- a--- a--- a--- a--- a--- a--- a--- a--- a--- a---
AH1  h 1.0e+0     AP2
```

```
AV1  v 1.0e+0     AP2
AH2  h 1.0e+0     AP3
AV2  v 1.0e+0     AP3
AH3  h 1.0e+0     AP4
AV3  v 1.0e+0     AP4
AH4  h 1.0e+0     BP5
AV4  v 1.0e+0     BP5
BH4  h 1.0e+0     BP3
BV4  v 1.0e+0     BP3
BH3  h 1.0e+0     BP1
BV3  v 1.0e+0     BP1
BH2  h 1.0e+0     BP1
BV2  v 1.0e+0     BP1
BH1  h 1.0e+0     AP1
BV1  v 1.0e+0     AP1
 $
*Group
Names are separated by blanks with up to 133 characters per line
Groups are separated by blank lines
AM BM
 $
*End
Emacs editing sequences follow:
Local Variables:
mode: text
tab-stop-list: (0 5 8 18 28 38 48 58 68 78 88 98 108 128)
indent-tabs-mode: nil
End:
```

2.7 Converting from Mad

Xorbit does not read the Mad input format, but it is a relatively easy matter to get the element-specification part of the Xorbit input from a Mad run. This is done by writing the parameters of the desired lattice in Mad (after Mad has computed it from its input) to a Mad OPTICS file, and then converting these parameters to Xorbit format using the program Madtox that is supplied with Xorbit.

Directions

1. Run Mad with the following lines in the input:

```
USE, MYLATTICE
SELECT, OPTICS, FULL
OPTICS, COLUMNS=NAME, KEYWORD, K0L, K1L, K2L, E1, E2, TILT, HKICK, VKICK
```

This will normally produce a file called optics. The name may be different on your system. It will be referred to as the *optics_file* below.

2. Run Madtox as follows:

```
madtox optics_file yourname.xin
```

The program will prompt you for whether you want Mad or Comfort sign conventions. Use the Mad convention. It will also prompt for the length of names in Xorbit. Pick this to be long enough to handle the names you use. (See NAMESIZE in the section on input switches.) If Mad can read your Comfort file, then you can use this same mechanism to convert Comfort input using Mad. In that case choose the Comfort sign convention in Madtox. Running Madtox without any arguments will return its usage.

The file, *yourname*.xin, that is produced is the element specification part of the input. It can be inserted into an input file below the NAMELIST input. (It includes the two alignment lines.) See the section on the input file. The name of the file does not have to have the extension .xin but it is recommended. It is the same sort of file that is produced by the File / Lattice File menu option.

The following caveats should be noted: Kickers should be HKICK and VKICK in Mad rather than multipoles. The last one or two digits of accuracy may be wrong owing to roundoff errors from the accuracy with which Mad prints the data in the OPTICS file. If this is important, it must be corrected by hand. Magnet errors must be put in by hand.

When the accuracy of the input in Mad and Xorbit is the same, the output has been found to be identical to the accuracy of the numbers in the Mad PRINT file.

3. Main Menu Bar

3.1 File Menu

Read Lattice File

Reads a file which is the same as the part of the input file that defines the lattice elements. The numbers in this file, except for the displacements, replace the current numbers for each element in the lattice. The lattice names, types, and multipole orders, must be the same as those read in via the input file. It is primarily a means of restoring a lattice saved with the **Write Lattice File** button. This button will bring up a file selection dialog box to choose the name of the file.

Write Lattice File

Creates a file which can be used as the part of the input file that defines the lattice elements or can be read in with the **Read Lattice File** button.. There is a choice of saving either the **Full** lattice (using the full names) or the lattice for a single **Sector** (using the reduced names). If used with the **Read Lattice File** button, the single-**Sector** option must be used. This button will bring up a file selection dialog box to choose the name of the file.

SDDS File

Creates a file in Self-Describing Data Set (SDDS) format with information about each element, including the name, type, multipole order, Twiss parameters, and orbit coordinates, as well as global parameters such as the tunes. There is a choice of **Create**'ing a new file or **Append**'ing to an old one. This button will bring up a file selection dialog box to choose the name of the file.

Quit

Terminates Xorbit.

3.2 Options Menu

Plots

Opens the plot window.

Show Tunes

Displays the horizontal and vertical tunes and total phase advances for the current configuration.

New Twiss

Causes new values for the Twiss parameters to be calculated. This is not always done automatically for efficiency reasons.

Restore Lattice

Restores the lattice as it was read from the input file. Does not restore correctors and does not restore bend parameters. Correctors are saved and restored separately as described just below. Bend parameters are not saved. If you wish to remember a set of bend parameter settings, you may leave an element information dialog box open for the bend in question. Since the parameters in the element selection dialog box are not updated, they can be used to restore the values at the time it was selected (or last changed).

Displacements

Controls the values for the displacements of the elements in the lattice. They can be **Save**'d, **Restore**'d, **Zero**'ed, or **New** random values can be calculated.

Correctors

Controls the corrector strengths. They can be **Save**'ed, **Restore**'d, or **Zero**'ed.

Channel Access

Controls Channel Access. There are two options: **Start** (or restart) Channel Access or toggle the **Echo**'ing of Channel Access messages. See the section on Channel Access.

Xvgr Plots

Starts Xvgr as an independent process to display several plots, including **Orbit** plots of **x**, **x'**, **y**, or **y'**; **Full Twiss** or single **Sector Twiss** plots of **Alpha**, **Beta**, or **Gamma**; and plots of strengths of **Correctors**, either **Horizontal** or **Vertical**. These plots may be printed, combined, or otherwise modified in Xvgr, which is a powerful plotting program. See the Xvgr instruction manual.

Xterm

Starts an Xterm window as an independent process. (It may be useful to run Mathematica or other programs in an Xterm window, since the Xterm window will remain after the other program has stopped, either inadvertently or on purpose.)

Devtest

Starts Devtest as an independent process in its own window. Devtest can be used to access Xorbit via Channel Access. See the instruction manual for Devtest.

Mathematica

Starts Mathematica as an independent process in its own window. Mathematica can be used to access Xorbit via Channel Access. See the instruction manual for Mathematica. There is a more detailed discussion of the Mathematica interface for Xorbit in the section on Channel Access.

3.3 Orbit Menu

Orbit Options

Brings up a dialog box with orbit options. See the section on dialog boxes.

Closed Orbit

Calculates the closed orbit for the current configuration.

Start Orbit

Starts a new orbit from the starting values given in the input file or modified in the orbit options dialog box. The number of turns is also specified as NTURNS in the input file or modified via the orbit options dialog box. The orbit starts with the INJ marker element.

Continue Orbit

Continues the orbit from its current values for another set of turns as specified as NTURNS in the input file or modified in the orbit options dialog box.

Statistics

Displays both the rms and maximum absolute values for all points on the orbit, the monitors only, the corrector strengths, and the quad displacements.

Save

Saves the current orbit associated with one of the possible save numbers.

Display

Displays one of the saved orbits on the graph area.

Difference

Sets the graph area to display the difference of all later orbits w.r.t the chosen one. Choosing **None** causes no differencing to occur (as is the case when Xorbit starts). The number of the orbit is shown at the right edge of the area above the graph area. If there is no differencing, then "True Orbit" is displayed. Note that choosing **Closed** differences the orbit w.r.t the current closed orbit, which may not be what is desired.

Girder-Corrected Orbit

Applies an algorithm that corrects the orbit by moving girders. The girders are specified in the input file. This algorithm is of limited interest, and it is better to do this sort of things via Channel Access using a program like Mathematica. See the section on correction algorithms, and see option blocks in the section on the input file.

Quad-Corrected Orbit

Applies an algorithm that corrects the orbit by moving quadrupoles. The positions of the quadrupoles that minimize the closed-orbit or the closed orbit at the monitors (depending on a switch in the input file) are calculated by a least-squares fit. See the section on correction algorithms, and see option blocks in the section on the input file.

First-Turn Orbit

Applies an algorithm, appropriate for the first turn, that corrects the orbit by adjusting each corrector in sequence (starting from the INJ marker) to make its associated monitor read zero. It can be chosen from the input file or the orbit options whether to do this for a single turn or whether to also iterate on the injection angle (at INJ) to make the orbit be closed. The correctors and the associated monitors are specified in the input file. It is often the case that the uncorrected orbit hits the wall. This is the most convenient (and realistic) way to get an initial closed orbit. See the section on correction algorithms, and see option blocks in the section on the input file.

3.4 Clear Menu

The Clear Menu appears both on the Main Menu Bar and on the Plot Menu Bar.

Clear

Clears all orbits in the graph area.

Redraw

Clears all orbits in the graph area and redraws the current orbit.

Update

Calculates and plots the orbit for the current parameters.

Auto Clear or **No Auto Clear**

Toggles between clearing the graph area after each new orbit is calculated and leaving the old orbits showing.

3.5 Help Menu

Xorbit uses HyperHelp from Bristol Technology to display its help. The program, HyperHelp, can be freely distributed with Xorbit and must be installed for this feature to work. The location of hyperhelp, the executable file, may be specified by the environment variable HHPATH if it is not in the standard place. The first time HyperHelp is accessed may take a short while. Once up, it may be left up or iconized, and access will then be much faster. There is a simple help file that can be displayed by the **Quick Notes** item in the event HyperHelp is not working. The help menu is nearly identical for both the main bar and the plot menu bar.

Contents

Displays the contents page for Xorbit help. You can jump to other topics by clicking on text that is green and underlined, or you can use the browse or search feature.

Overview

Displays the overview page. (Appears on the main window Help.)

Plot Window

Displays the Plot Window page. (Appears on the plot window Help.)

Using HyperHelp

Displays help on using HyperHelp. The location of the hoh.hlp file may be specified by the environment variable HOHPATH if it is not in the standard place.

Quick Notes

Displays a short overview of Xorbit in a dialog box. This feature does not require HyperHelp.

Version

Displays the Xorbit version. This feature does not require HyperHelp.

4. Plot Menu Bar

4.1 Options Menu

Plot Options

Brings up a dialog box with plot options. See the section on dialog boxes.

Design Mode

Brings up a dialog box that implements the design mode. In this mode the element strengths and the drift spaces can be changed by clicking with the mouse or by entering values in a text box. The plots in the plot window change as the elements are changed, allowing immediate visual feedback of the results of the changes on any of the above parameters. See the section on dialog boxes.

Remove

Removes the plot window.

Quit

Terminates Xorbit.

4.2 Clear Menu

The Clear Menu is the same as for the main menu bar. See that section.

4.3 Help Menu

The Help Menu is nearly the same as for the main menu bar. See that section.

5. Dialog Boxes

5.1 Orbit options

The controls in this dialog box are used to customize how the orbit is displayed in the graph area and to set other parameters that are related to the orbit.

Top Controls

Remove

Removes this dialog box.

Initial Twiss

For beam lines brings up a dialog box that allows changing the initial Twiss parameters. (See Initial Twiss Parameters Dialog Box.)

Help

Provides help on this dialog box.

Plot:

These controls determined how the orbit is displayed.

x

Toggle display of the x coordinate of the orbit.

y

Toggle display of the y coordinate of the orbit.

x-BPM

Toggle display of markers at x positions of the orbit where there are monitors.

y-BPM

Toggle display of markers at the y positions of the orbit where there are monitors.

x-err

Toggle display of markers at the x positions of non-zero element displacements.

y-err

Toggle display of markers at the y positions of non-zero element displacements.

Symbols

Toggle display of symbols representing the magnet elements on the s axis. Bends are one unit high, and edges are a half unit higher. Quads are two units high and are above or below the axis depending on whether they are focusing or defocusing. Multipoles of order n are n units high and are above or below the axis depending on the sign of the larger of a or b. The injection point is three units high. Monitors are small, square boxes.

BPM-only

Toggle display of the orbit at all points or only at monitor points. Showing the orbit with BPM Only and x-BPM and/or y-BPM represents what is known in a real device.

BPM Only:

These controls determine options for when BPM Only above is selected.

True Values

Shows the actual coordinates.

Values as Read

Shows the coordinates as actually read by the monitors. The offsets (errors or displacements) are subtracted from the actual values.

Start Orbit From:

These controls determine characteristics of the orbit that is started or continued via the Orbit Menu.

Specified

Use the x0, x0', y0, and y0' values specified below as initial values at the injection point when starting the orbit.

FTO

Use the x0, x0', y0, and y0' values from the last-calculated first-turn orbit as initial values at the injection point when starting the orbit.

CLO

Use the x0, x0', y0, and y0' values from the last-calculated closed orbit as initial values at the injection point when starting the orbit.

x0:, x0':, y0:, and y0':

These text boxes set the initial values at the injection point which are used when starting the orbit if Specified above is selected. See Orbit Menu for how to start or continue the orbit.

Orbit Turns:

This text box sets the number of turns to take when the orbit is started or continued. See Orbit Menu for how to start or continue the orbit.

Xvgr Plot:

These controls determine if Xvgr plot files are created. These files can be read into Xvgr, then viewed, printed, and modified. Xvgr plots can be also displayed from Xorbit via the Options Menu.

ORB

Make Xvgr plot files when the orbit is calculated. These files have names starting with the prefix of the input file name and ending with the extensions: .porbx, .porby, .porbxp, and .porbyp.

FTO

Make Xvgr plot files when the first-turn orbit is calculated. These

files have names starting with the prefix of the input file name and ending with the extensions: .pftox, .pftoy, .pftoxp, and .pftop.

CLO

Make Xvgr plot files when the closed orbit is calculated. These files have names starting with the prefix of the input file name and ending with the extensions: .pclox, .pcloy, .pcloxp, and .pcloyp.

First Turn:

These controls determine which of two ways the first-turn orbit is calculated. See the section on correction algorithms.

Correct Only

Correct the orbit for one turn only.

Correct and Close

Correct the orbit for one turn and iterate on the injection angle until the orbit is closed.

5.2 Plot Options

Top Controls

Remove

Removes this dialog box.

Redraw

Clears all orbits in the plot graph area and redraws the current orbit.

Update

Calculates and plots the orbit for the current parameters.

Help

Provides help on this dialog box.

Options:

Immediate Twiss

Toggles whether the Twiss parameters are calculated immediately after a change in the element parameters or not. If not, they may be calculated via the Options Menu when desired. For visual feedback in the design mode, it is usually better to set this option on.

Symbols

Toggle display of symbols representing the magnet elements on the s axis. Bends are one unit high, and edges are a half unit higher. Quads are two units high and are above or below the axis depending on whether they are focusing or defocusing. Multipoles of order n are n units high and are above or below the axis depending on the sign of the larger of a or b. The injection point is three units high. Monitors are small, square boxes.

Table Controls

For each parameter in the table, these controls determine:

Column 1: Whether this parameter is plotted in the plot graph area or not. There are also input switches, IPLOT*xxx*, that control whether a variable is plotted or not.

Column 2: The scale factor by which the values for this parameter is multiplied before plotting. This is one of two ways to control the relative sizes of the curves in the plot graph area. The other is the upper and lower limits control above the plot graph area. There are also input switches, SCALE*xxx*, that specify the scale for a variable.

Column 3: The color of the curve for this parameter. Any name which can be used as a color resource name may be used. These include colors in the color database (usually /usr/lib/X11/rgb.txt) and names in the forms #RGB, #RRGGBB, #RRRGGGBBB, and #RRRRGGGGBBBB, where R, G, and B are single hexadecimal digits representing the amounts of red, green, and blue. (For a 256-color system two digits for each color are appropriate and Red is #FF000, for example.)

5.3 Design Mode

This dialog box allows the easy changing of elements with immediate feedback in the plot graph area. Typically this mode would be used for designing a lattice by changing element strengths and positions while watching their effect on the beta functions and dispersion.

The current element is chosen by clicking on its s value in either the graph area or the plot graph area or by clicking its icon in the icon bar.

Top Controls

Remove

Removes this dialog box.

Redraw

Clears all orbits in the plot graph area and redraws the current orbit.

Update

Calculates and plots the orbit for the current parameters.

Restore Lattice

Restores the lattice as it was read from the input file. Does not restore correctors and does not restore bend parameters. Correctors are saved and restored separately as described just below. Bend parameters are not saved. If you wish to remember a set of bend parameter settings, you may leave an element information dialog box open for the bend in question. Since the parameters in the element selection dialog box are not updated, they can be used to restore the values at the time it was selected (or last changed).

Help

Provides help on this dialog box.

Reset Current Value

Resets the value in the Current Value box to the true current value for the element. The arrow keys increment from the value shown in the Current Value box. This value is only updated for changes made via the arrow keys. Consequently, a reset to the true value may be useful after restoring the lattice or making other changes not using the design mode dialog box. Alternatively, the element can easily be restored to the value before the Restore Lattice by typing <CR> in the current value box instead of resetting the current value. It is also possible to keep track of previous settings by using the Element Information dialog boxes.

Options:

All Sectors

Specifies whether the changes apply to the same element in all sectors or just to this particular element.

Attach To:

Specifies which strength is associated with the value that is changing. See the section on element input.

A

The value that changes is the A strength.

B

The value that changes is the B strength.

Position Change:

Specifies the increment for position changes in meters. The modifications are not implemented until <CR> is pressed.

Value Change:

Specifies the increment for strength changes in the units in which the strengths are specified. The modifications are not implemented until <CR> is pressed.

Current Value:

Displays the current strength of the selected element. The strength can be changed via the text box. The modifications are not implemented until <CR> is pressed.

Current Element:

Displays the name and index in the lattice of the current element. This text box is for display only and cannot be edited.

Arrow Keys

The left and right arrows move the element sideways. It cannot cross the adjacent element, however. The up and down arrows increment the strength.

5.4 Element Information

This dialog box displays information about an individual element and allows its strengths and displacements to be changed (if that is appropriate).

The information is current at the time the dialog box is displayed. Except for values that are changed via the text boxes, this information is not updated as the orbit changes. One can keep track of what is happening at an element by displaying new boxes as changes are made to the orbit.

The information includes the type of the element, its name and index in the lattice, the current x, x', y, and y' coordinates of the orbit; the s coordinate of the element (in the current units); the horizontal and vertical Twiss parameters, α_h, β_h, γ_h, α_v, β_v, and γ_v; the horizontal and vertical phases, φ_h and φ_v (relative to the injection point in units of π); and the horizontal dispersion, η_x, and its derivative, η'_x. Both x and h refer to horizontal, and both y and v refer to vertical.

There is a plot of the phase ellipses, red for horizontal and blue for vertical.

When appropriate there are text boxes that can be used to change the A and B strengths and the x and y displacements, x-error and y-error. (See the section on element input The modifications are not implemented until <CR> is pressed.

There is a text box at the bottom for writing notes. (Nothing is done with what is written.) This facilitates keeping track of changes in the element parameters as the orbit changes.

Top Controls

Remove

Removes this dialog box.

Remove All

Removes all the element dialog boxes that are currently displayed.

Help

Provides help on this dialog box.

5.5 Initial Twiss

This dialog box allows initial values, α_{h0}, β_{h0}, α_{v0}, and β_{v0} for the Twiss parameters and η_{x0} and η'_{x0} for

the dispersion, to be set when there is a beam line. These represent the values at the injection point. The modifications are not implemented until <CR> is pressed.

Top Controls

Remove

Removes this dialog box.

Help

Provides help on this dialog box.

6. Input Switches

6.1 General

Switch	Default	Description
IDISTCHECK	0	1,0: Do not, do check random number distribution
IECHOIN	0	1,0: Do not, do echo input file
IECHOSW	0	1,0: Do not, do echo input NAMELIST
IINTERACTIVE	0	1,0: Do not, do use interactive, GUI interface
NAMEPREFIX	'X'	Prefix for full names, used for INAMECONV=1,2
NAMESIZE	4	1-8: Length of element names in input file
INAMECONV	0	0: Full names are the same as short names (those entered in the input file) 1: Full names are nameprefix+sector number+shortname, *e.g*. X7Q1 2: Full names are nameprefix+sector number:shortname, *e.g*. X7:Q1
IPLCS	0	0: Do not plot Courant Snyder (Twiss) parameters 1: Plot for a single (the first) sector 2: Plot for the full lattice
IRING	1	0: Treat lattice as a beam line 1: Treat lattice as a ring
xxx0 *e.g*. ALPHA0	0.0	Initial values for the variable xxx when IRING=0 (xxx may be any of : ALPHAH, ALPHAV, BETAH, BETAV, ETAX, or ETAXP.)
ISEED	0	Number of sets of random numbers of size ISEEDSKIP to initially discard
ISEEDSKIP	1000	Size of random number sets that are initially discarded
ISUNITS	0	0: s units are meters n: s units are 1/n *th* of a sector (n > 1 may be used to divide a non-symmetric lattice into sectors)
NSECT	40	Number of identical sectors

6.2 Channel Access

Switch	Default	Description
ICAINIT	0	1,0: Do not, do initialize channel access on startup
ICAECHO	0	1,0: Do not, do echo channel access messages

ICAWAIT	0	0: Do not wait for channel access messages 1: Only wait for channel access messages 2: Do first-turn correction, then only wait for channel access messages (Only meaningful for non-interactive mode)

6.3 Correctors

Switch	Default	Description
ICORSORT	0	0: Do not print corrector strengths 1: Print in lattice order 2: Print in ascending order 3: Print in descending order.
IPLCORH	0	1,0: Do not, do plot horizontal corrector strengths in <inputname>.pcorh
IPLCORV	0	1,0: Do not, do plot vertical corrector strengths in <inputname>.pcorv

6.4 Orbit

Switch	Default	Description
NTURNS	0	0: Do not do orbit >0: Number of turns for orbit
ICLO	0	0,1: Do not, do find closed orbit
IFTO	0	0,1,2: Do not, do find first-turn orbit, do find closed first-turn orbit
IORB	0	0,1: Do not, do calculate orbit from starting values for NTURNS turns
IPLCLO	0	0,1: Do not, do plot closed orbit in <inputname>.pclox, xp, y, yp
IPLFTO	0	0,1: Do not, do plot first-turn orbit in <inputname>.pftox, xp, y, yp
IPLORB	0	0,1: Do not, do plot orbit in <inputname>.porbx, xp, y, yp
XSTART	0.0	Starting value of x for orbit (m)
YSTART	0.0	Starting value of y for orbit (m)
XPSTART	0.0	Starting value of x' for orbit (radians)
YPSTART	0.0	Starting value of y' for orbit (radians)
XWALL	0.01	Value of x where orbit is considered lost (m)
YWALL	0.01	Value of y where orbit is considered lost (m)

6.5 Plots

Switch	Default	Description
ITWISSUPDATE	1	0,1: Do not, do update the graph areas when Twiss parameters are calculated (because element parameters have changed) when the plot window is showing
IPLOTxxx	0	0,1: Do not, do plot the variable xxx in the plot window (xxx may be any of: ALPHAH, ALPHAV, BETAH, BETAV, GAMMAH, GAMMAV, X, Y, XP, YP, ETAX, or ETAXP.)
SCALExxx	1.0	Scale the values for variable xxx in the plot window (xxx may be any of: ALPHAH, ALPHAV, BETAH, BETAV, GAMMAH, GAMMAV, X, Y, XP, YP, ETAX, or ETAXP.)

7. Option Blocks

Xorbit is designed so that input for new options can be added easily in blocks. The start of each block is specified by a line containing "*name" and ends with a line that contains " $". (The $ is in column 1 or 2.) The order of the option blocks is not important. Currently the possible name's are:

group:	Grouping for elements with common power supplies
firstturn:	First-Turn Correction Algorithm
girdercorrected:	Girder Correction Algorithm
quadmovement:	Quad-Movement Correction Algorithm

(Case and spaces are ignored in name.) See the sample input to see how options blocks are included in the input file. The details of each type of option block are described in the following subsections.

7.1 Grouping of Elements

Elements can be grouped so that a change to the parameters of any one will also be applied to the others in the group. This feature is useful, for example, for magnets that are driven by a common power supply. The first two lines in the option block are ignored. After these lines, the elements can be grouped by specifying their names, separated by blank spaces or commas, on successive lines. A blank line separates groups from each other. The groups will be numbered, starting from 1, in the order in which thay appear. The following is an example of a group option block:

```
*Group
Names are separated by blanks with up to 133 characters per line
Groups are separated by blank lines
P1B1 P2B2 P3B3 P4B4
P1B2 P2B2 P3B2 P4B2

P1Q1 P2Q1 P3Q1 P4Q1

P1Q2 P2Q2 P3Q2 P4Q2

P1Q3 P2Q3 P3Q3 P4Q3

P1Q4 P2Q4 P3Q4 P4Q4
 $
```

In this example, the eight bending magnets with names ending in B1 or B2 are grouped in the first group. There are four more groups, one for each of the quadrupoles with names ending in Q1, Q2, Q3, and Q4.

7.2 First-Turn Correction

The first-turn correction is a way of correcting the orbit by associating each corrector with one or more monitors. The correction is performed by setting the first corrector after the injection point sucessively to two different values and looking at the resulting average of the monitor readings of the associated monitors. The corrector setting is then linearly iterated using a Newton-Rhapson algorithm until the monitor average is sufficiently close to zero. This process is repeated corrector by corector around the ring or to the end of a line.

For a ring there must be a monitor just before the injection point for it to be possible to do multiple sectors. This requirement may define the placement of the injection point. The algorithm may be applied to a single turn (in which case the orbit is not guaranteed to be closed) or a Newton-Raphson iteration may be made on

the injection angle after each single-turn calculation until the orbit closes. This option is controlled by the input switch, IFTO, or the Plot / Plot Options... dialog box.

The corrector-monitor associations are specified with the first-turn option block. The first two lines are ignored and may be used for spacing information. Each line after the first two specifies which monitors are associated with a single corrector, the plane (“h” or “v”) of the correction, and the maximum absolute corrector strength. The format is:

corrector, plane, maximum-strength, monitor1, monitor2, ...

FORMAT(A<namesize>,1X,A1,A1,1X, F10.0,<MM>(1x,a<namesize>))

The parameter MM, representing the number of monitors for each corrector, is defined in fturncom.h. By default it is 2. The following is an example of a group option block:

```
*First turn
name t  strength  bpm1 bpm2 bpm3 bpm4 bpm5 bpm6 bpm7 bpm8 bpm9 bpm10 ...
a--- a f--------- a--- a--- a--- a--- a--- a--- a--- a--- a--- a---
AH1  h 1.0e+0     AP2
AV1  v 1.0e+0     AP2
AH2  h 1.0e+0     AP3
AV2  v 1.0e+0     AP3
AH3  h 1.0e+0     AP4
AV3  v 1.0e+0     AP4
AH4  h 1.0e+0     BP5
AV4  v 1.0e+0     BP5
BH4  h 1.0e+0     BP3
BV4  v 1.0e+0     BP3
BH3  h 1.0e+0     BP1
BV3  v 1.0e+0     BP1
BH2  h 1.0e+0     BP1
BV2  v 1.0e+0     BP1
BH1  h 1.0e+0     AP1
BV1  v 1.0e+0     AP1
 $
```

There is a difference between an n1-multipole and a corrector. Only the n1-multipoles that appear in the first-turn option block are correctors. Their strengths, which vary from sector to sector are kept track of separately, and they can be saved and restored independently of the other elements.

7.3 Quadrupole-Movement Correction

7.4 Girder-Movement Correction

8. Channel Access

8.1 How Channel Access is Emulated

Xorbit provides a Channel Access capability which emulates that used in the APS control system. The real (Epics) Channel Access communicates with real devices over the Ethernet. Xorbit Channel Access communicates with Xorbit, which simulates many aspects of the real devices, via Unix interprocess communication, primarily via a message queue.

Channel-access interfaces to such programs as Mathematica, Devtest, WingZ, and Excel work in just the same way with Xorbit as they do with the real machine. The only necessary change is that they need to be linked with the Xorbit Channel Access routines rather than the Epics Channel Access routines. The Xorbit routines have the same names and arguments and, in fact, use the same header file, cadef.h.

Apart from emulating the real Channel Access, these routines provide a powerful and easily modified way to perform complex procedures, such as orbit correction, on the lattice in Xorbit. They also make the extensive features, such as plotting or matrix manipulation, in programs such as Mathematica available for analyzing orbit dynamics.

Devices are read or set via their process variable names. The process variable names used in Xorbit are of the form:

elementname.variable

e*lementname* is the full name of the element in Xorbit, e.g., S34AQ1. The possible values for *variable* include:

x	x value
y	y value
xp	x' value
yp	y' value
a	A strength
b	B strength
s	s
alphah	α_h
alphav	α_v
betah	β_h
betav	β_v
gammah	γ_h
gammav	γ_v
psih	ψ_h
psiv	ψ_v
bpmx	x reading
bpmy	y reading
nuh	ν_h
nuv	ν_v
l	element length
xerr	x displacement
yerr	y displacement
etax	η_x
etaxp	η'_x
e1	E1 edge angle
e2	E2 edge angle
fint	FINT edge parameter
hgap	HGAP edge parameter

To read the x displacement of S34AQ1 one would use the process variable name S34AQ1.xerr, and to set the strength of horizontal corrector S1AH2 to, say, 0.1 mrad, one would use the process variable name S1AH2.a. In Mathematica, for example, this might be done by

```
xdisp=CaGet["S34AQ1.xerr"]
```

and

```
newstrength=CaSet["S1AH2.a",.0001],
```

respectively. In a spreadsheet it might be done by selecting the cell or cells with the appropriate name or names and indicating the cells for the input and output values before running an appropriate macro.

Channel-access interfaces for Mathematica, Devtest, WingZ, and Excel have been written at the Advanced Photon Source. These are separate programs from Xorbit. See the instruction manuals for the relevant program interfaces for details on how to use the interfaces, and see the instruction manuals for the programs themselves for how to use the programs.

8.2 Mathematica Interface to Channel Access

The program CaMath is supplied with Xorbit to provide an interface that allows using the power of Mathematica with the orbit tracking capability of Xorbit. This program contains a subset of the Mathematica channel-access capability developed for the Advanced Photon Source as mentioned above, but it is sufficient to utilize Xorbit calculations from within Mathematica using the variables described above. (Most of the routines in CaMath were written by Ben-Chin Cha.) This discussion assumes that you are familiar with Mathematica and how to run it on your system.

Xorbit can be run with channel access either interactively or non-interactively. Non-interactively, it just waits for Channel-Access messages. Interactively, you have access to any of the features of the Xorbit GUI while you are running Mathematica.

The interface between Mathematica and Xorbit is established by connecting to CaMath via sockets. This is done using the Mathematica Install["*command*"]. If the *command* is the CaMath program itself (with the proper path), everything else is done for you. This is the easiest way if Mathematica is on the same machine as Xorbit. Otherwise, you can start CaMath independently. It will give you a socket name, which you can then use for *command* when using Install["*command*"] from within Mathematica. This method seems to be more solid if there are problems. After you have established a connection with Install[], you have two additional Mathematica commands, CaGet[] and CaSet[], which are described below. See the Mathematica manual for more details on the MathLink interface, if necessary.

Directions

More precisely, the steps to run Xorbit with Mathematica are as follows. (Optional steps are enclosed in [ ]'s, and the full path name for CaMath is assumed to be ~/src/camath/CaMath):

1. Start Xorbit:

interactive: Set IINTERACTIVE=1 in the input file. Set ICAWAIT=1 or use the Channel Access / Start option in the Options menu.

non-interactive: Set IINTERACTIVE=0 and ICAWAIT=1 in the input file.

```
xorbit <name> &                 [Start Xorbit]
```

2. Run Mathematica:

```
math                            [Start Mathematica]
Install["~/src/camath/CaMath"]  [Start the interface from within Mathematica]
...                             [Do whatever you want to do]
Quit                            [Terminate Mathematica]
```

or

```
~/src/camath/CaMath          [Run CaMath from outside Mathematica]
        <CR>                 [Get <socket> to use below]
        [^Z]                 [Suspend if running from a single tty]
        [bg]                 [Put CaMath in the background]
[rlogin <machine>]           [If you want to use another machine]
math                         [Start Mathematica]
        Install["<socket>"]  [Connect to the socket returned from CaMath]
        ...                  [Do whatever you want to do]
        Quit                 [Terminate Mathematica]
[logout]                     [If using another machine]
```

3. Terminate Xorbit:

interactive: Use "Quit" button.

non-interactive: Xorbit traps SIGTERM and SIGINT in order to terminate cleanly. There are several ways to utilize this feature:

```
        kill %xorbit     [Kill the Xorbit process]
or
        fg %xorbit       [Bring the process to the foreground]
        ^C               [Kill it]
or
        ps | grep xorbit [Get the process id, <pid>]
        kill <pid>       [Kill it]
```

CaGet and CaSet

These two functions are all that are needed to access any of the channel access variables from Xorbit. Their syntax is:

values=CaGet[{s1, s2,...}] or values=CaGet[s1, s2, ...]

newvalues=CaSet[{s1, s2,...},{val1, val2, ...}] or newvalues=CaGet[s1, s2, ... ,val1, val2, ...]

The quantities s1, s2, ... are Mathematica strings which are the process variable names (see above), and the quantities, val1, val2, ... are the numbers to which these process variables are to be set in the case of CaSet. The return quantities, values or newvalues, will be lists with the same length as {s1, s2, ...}, and are the current settings. The second form (without lists) is more convenient when using only a single process variable.

For example the following Mathematica lines would increment two correctors, S1AH1 and S1AH2, by 0.1 mrad.:

```
names={"S1Ah1.a","S1AH2.a"}

oldvals=CaGet[names]

newvals=CaSet[names,oldvals+.0001]
```

The file xorbit.m contains a number of typical Mathematica procedures, which should be usable with any machine to do involved procedures such as measuring response matrices or performing orbit correction.

8.3 Technical Details

The real Channel Access uses the Ethernet to communicate with real devices in the machine. Xorbit Channel Access uses Unix interprocess communication (IPC), in particular signals and a message queue, to communicate with data and processes in Xorbit. When Channel Access is started in Xorbit, either on startup if ICAINIT=1 or by choosing the Channel Access / Start option in the Options menu:

1. A file, /tmp/xorbit.channel, is created. This file is used as a key for other process to identify the Xorbit channel access mechanism.

2. A message queue for channel access messages is initialized and flushed of any old messages.

3. A message is left in the queue identifying the process id of Xorbit.

4. Xorbit begins to trap the signal SIGUSR1.

5. Xorbit goes on about its business.

When another process, such as Mathematica, initializes:

1. It gets the key from the file.

2. It looks for the Xorbit message on the queue and gets the Xorbit process id if the message is there.

When another process wants information from Xorbit:

1. It flushes any old messages to itself.

2. It puts a message on the queue stating what it wants. The message contains its process id.

3. It sends a SIGUSR1 signal to Xorbit, using the Xorbit process id.

4. It waits for a reply with a timeout in case none comes.

5. When the reply comes, it checks that it is valid and then processes it.

When Xorbit receives a SIGUSR1 signal:

1. It stops what it is doing and looks in the queue for a message.

2. Depending on the nature of the message, it performs the actions that are required, and sends a reply with the required information.

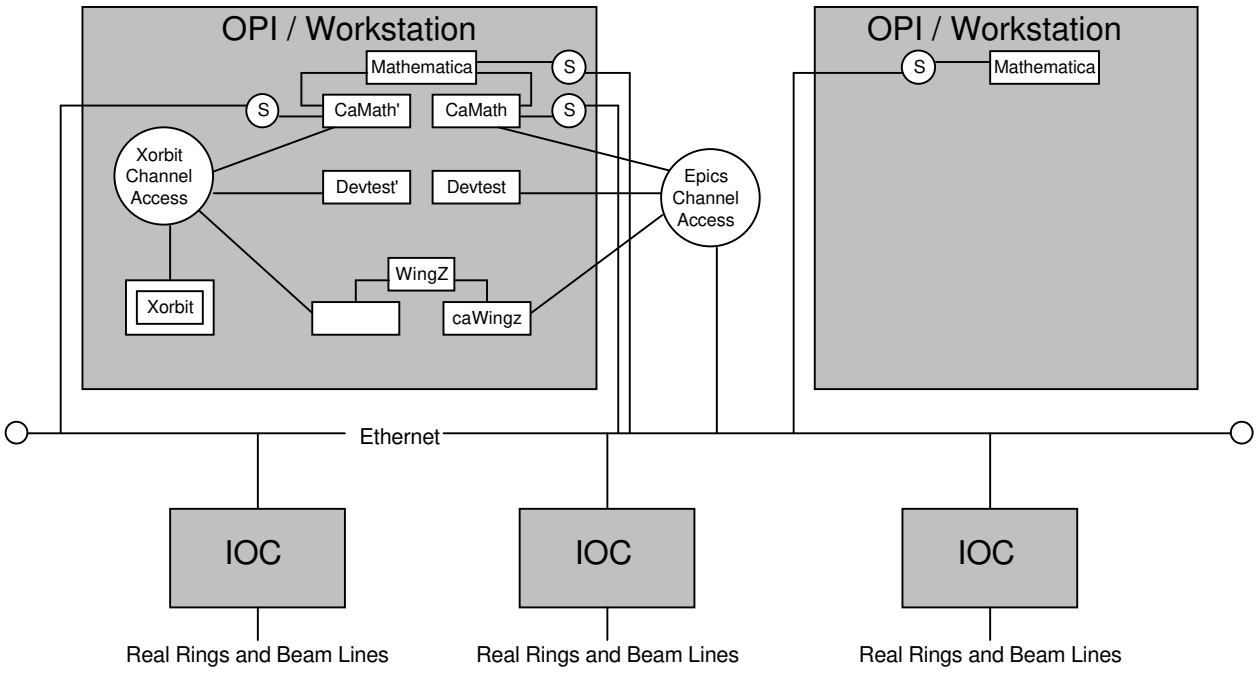
3. It continues getting messages from the queue and replying to them until the queue is empty.

4. It calculates new Twiss parameters and a new closed orbit if necessary. The closed orbit calculation may take a few seconds, depending on the complexity of the lattice. (Think of this time as the damping time.) It is better to insure no new messages are sent during the closed orbit calculation..

5. It returns to what it was doing.

The message traffic can be viewed from Xorbit by setting ICAECHO=1 or by choosing Channel Access / Echo from the Options menu.

For Xorbit Channel Access to work with another application, that application needs to have an interface that uses the Xorbit Channel Access routines (which are located in the file ca.c) by linking these routines with the interface. The Mathematica interface described above is an example of such an interface. There are others that have been developed at the Advanced Photon Source. The routines in ca.c have been written so that the interface could use the real (Epics) Channel Access just by using the Epics Channel Access routines instead of those in ca.c. The diagram illustrates schematically the relations between Xorbit, these interfaces (*e.g.* CaMath), and the corresponding programs (*e.g.* Mathematica) for both Xorbit Channel Access and the Epics Channel Access. The interfaces to Xorbit are denoted with a prime ('), and S stands for a Unix socket. IOC stands for Input-Output Controller, and OPI stands for Operator Interface.



Schematic of Xorbit and Epics Channel Access

9. Description Of The Code

Xorbit is a mixture of Fortran and C routines. Most of the accelerator physics calculations are written in Fortran, and the GUI interface is written in C. It utilizes Motif Version 1.1 for the interface. It has been developed and used only on Sun workstations.

Because of the mixed language and the use of Unix interprocess communication for the Channel Access capability, it is not expected to be immediately portable to other platforms, since both of these features are machine dependent. Undoubtedly changes will have to be made as Sun migrates to Solaris.

It should be noted that the Sun-supported GUI interface for Sun workstations is Open Look, not Motif. This has caused problems, to the extent that with Open Windows 2 and the Olwm window manager, Sun's previous implementation of Open Look, the Motif interface simply did not work well enough to be used. There have been few problems with Open Windows 3, however, and the new Olwm. The program has always worked well, as could be expected, with the Mwm window manager, which is Motif based. The reason for using Motif was that the Advanced Photon Source control system is Motif-based. (At this time it has not been determined what brand of workstations will be used for the APS, but almost all workstations are Sun's at present.)

Xorbit makes use of several other programs, such as Xvgr, Mathematica, Devtest, Awe, and Xterm. It can be compiled to avoid reference to any of these programs if desired. Such changes can be made in the header file, sitespecific.h. This file also contains pathnames and command lines that may be different on different machines. Other site-specific information appears in the makefile.

Xorbit can be either single or double precision. The Fortran source is written treating the variables as REAL*4. They will be REAL*8 if the program is compiled with Sun's f77 using the options, -r8 -i4. The C source is written allocating floating-point variables as REAL. If DOUBLE is not defined, REAL gets defined as float in xorbit.h, otherwise it is defined as double. One way to define DOUBLE is to include the option, -DDOUBLE, on the cc comand line. The code seems to work well in single precision. Double precision is, of course, more accurate and safer. Double precision is suggested.

Files

The Xorbit code consists of the following files. (The minimum set of files necessary to run Xorbit are xorbit, EvansApp, xorbit.hlp, and an input file):

Executable file

xorbit

Applications Default Files

EvansApp
HyperHelp

(These files specify necessary resources for Xorbit and suggested resources for HyperHelp. They should usually be put in your home directory or in /usr/lib/X11/app-defaults. Put them in your home directory if you are in doubt. These files are standard files for X Windows applications and may be used to customize the applications if desired. There are a number of other places in which these files may be located, and there are a number of other ways to specify the resources. See your favorite X Windows manual for more information. Xorbit will not work well if these files are missing from one of the proper places.)

Help file

xorbit.hlp

If the environment variable XORBITHOME is set, Xorbit expects to find this file in $XORBITHOME/xorbit.hlp. Otherwise, the expected location of this file is hard coded in sitespecific.h. If there is a specific pathname specified in sitespecific.h, then Xorbit will look for the file there. If it is specified in sitespecific.h as "xorbit.hlp", making it independent of the location of the xorbit files, then the file must be in your current working directory or a link made in your current working directory to its standard location. The link method saves disk space and can be done with the Unix command:

ln -s <fullpath>/xorbit.help xorbit.hlp

Makefile

makefile	Builds the program using the Unix Make utility

(The Motif libraries and include files must be available.)

Fortran Source Files

fturn0.f
fturn1.f
fturn2.f
fturn3.f
fturn4.f
fturn5.f
fturn6.f

C Source Files

xorbit.c	Main program, most of the interface
pa.c	Plot area routines
io.c	Input/Output routines
help.c	Help routine for HyperHelp

Fortran Include File

fturncom.h	Common blocks

C Include Files

xorbit.h	Main include file, Function prototypes, Global variables
xorbitcom.h	Defines the C structures that correspond to the Fortran common blocks
sitespecific.h	Site-specific definitions
help.h	Header for HyperHelp routines
awe.h	Header for Awe routines
ca.h	Header for Channel Access routines
cadef.h	Header for Epics Channel Access routines
caerr.h	Included by cadef.h
caeventmask.h	Included by cadef.h
db_access.h	Included by cadef.h
db_addr.h	Included by cadef.h
lstLib.h	Included by cadef.h

Icons and Bitmaps

aps.icon
bend.icon
blank.icon
dquad.icon
drift.icon
fquad.icon
hcor.icon
inj.icon
ledge.icon
monitor.icon
redge.icon
sext.icon
thin.icon
vcor.icon
xbpm.icon
xerr.icon
ybpm.icon

yerr.icon

(There are also files with the same prefix and extension .xpm instead of .icon that are used in the HyperHelp package, which requires a different format for bitmaps)

HyperHelp Files

xorbit.hlp	Compiled help file that the HyperHelp viewer uses
xorbit.hpj	Help project file
xorbit.hht	Help text file in .hht format
input.hht	Help text file in .hht format
xorbit.help.txt	Simple text help file

(The HyperHelp compiler from Bristol Technologies must be purchased to compile xorbit.hlp from the other files. The compiled file, xorbit.hlp, is all that is necessary to use the HyperHelp viewer, however, and the viewer *may* be distributed with Xorbit.)

Sample Input

aps.in	Input for the APS storage ring

(Input files for other machines may be available as well.)

Auxilliary Programs

madtox	Converts Mad OPTICS files to Xorbit input format
madtox.f	Fortran source for Madtox
CaMath	Mathematica interface
CaMath.c	Source for CaMath
cafunc.c	Source for CaMath
hash3.c	Source for CaMath

(The sources for CaMath were mostly written by Ben-Chin Cha. There is also a makefile, and the CaMath files should be in their own directory. MathLink routines from Mathematica are necessary to build CaMath.)

Channel Access Routines

ca.c	Xorbit Channel Access routines

(These routines are linked with an interface to Channel Access. They replace the Epics Channel Access routines so that the interface is with Xorbit rather than with the APS control system via the Ethernet. The header file, ca.h, as well as the Epics header, cadef.h, are used.)

Other Independent Programs Used by Xorbit

awe	Awe program (Mike Borland, APS)
devtest	Devtest (Ben-Chin Cha, APS)
hyperhelp	HyperHelp viewer (Bristol Technology, Inc., may be distributed with Xorbit)
math	Mathematica (Wolfram Research, Inc.)
mpl	Mpl Toolkit (Mike Borland, APS)
xterm	Xterm (comes with the X distribution)
xvgr	Xvgr (Paul Turner, FTP from amb4.ese.ogi.edu)

(These programs need to be obtained from their respective sources.)