Running Shadow under UNIX at ESRF

Manuel Sanchez del Rio
European Synchrotron Radiation Facility
BP 220, 38043 Grenoble Cedex (France)

E-mail: srio@esrf.fr
Report: EXP/MSR/91/08

May 30, 1991

Revised: October 1992

In this manual the special features of the SHADOW version installed in the UNIX machines at the ESRF are explained. It is not a tutorial, primer or autoconsistent document, but it gives important ideas and tricks to run easier SHADOW for the ESRF beamlines.

Contents

1 Introduction. Why a UNIX version of SHADOW? **1**

2 Setting and running SHADOW. Differences with VMS version **1**

3 Post-Processes. Results analysis **2**

4 Analysis with IDL **3**

5 Future Developments **5**

6 Where to find more **6**
6.1 Reference 6
6.2 Papers 6

A Shadow variables **7**
A.1 Column description 7
A.2 Glossary 7
A.3 Source description 8
A.4 Optical Element description 9

B IDL utilities **13**
B.1 Basic utilities 13
B.2 Advanced utilities 15
B.3 Tools 15

1 Introduction. Why a UNIX version of SHADOW?

SHADOW is the most extended code for ray-tracing simulation of an optical system, specially optimized for beamline design. SHADOW is a set of FORTRAN routines developed by F. Cerrina and coworkers at Center for X-ray Lithography (University of Wisconsin), in continual upgrading and development. It is used by the most important synchrotrons around the world and its use is growing continously.

The philosophy of the program is the generation of a light source consisting of a finite number of rays with distribution following the desired characteristics (dimensions, divergences, energy range, polarization), and passing each ray through an optical system. In fact SHADOW is extremely powerful for its ability to simulate synchrotron radiation sources (bending magnet, wigglers and undulators). This simulations are based on a Monte Carlo random generation of the rays which follow a probability distribution calculated with the machine parameters. After the source is generated, an optical system consisting of a number of mirrors, filters, slits, perfect or imperfect crystals, gratings, etc. is defined. SHADOW calculates for each ray the intercept, incident and reflected directions, absorption, etc. on each optical element along the optical system or beamline.

Although SHADOW is a Geometrical Optics based program, different models of Physical Optics are implemented for calculating reflectivities of optical elements , for instance, Fresnel Formulae for mirrors or Dynamical Theory of Diffraction for perfect crystal calculations. After tracing the optical system, a set of satellite programs are used for analyzing the output files. These programs permit the extraction of the output intensities, energy resolution, geometry of the image, phase space diagrams, etc. Other utilities help the user to define beamline parameters, such as apertures, curvature radii, etc.

The increasing use of UNIX computers for scientific usage has forced the SHADOW developers to adapt the code to these operation systems. UNIX workstations present higher speed, lower prices, more independence and better graphic interfaces than the big shared VMS mainframes.

SHADOW has recently been installed in the ESRF UNIX network in order to facilitate calculations for ESRF beamlines. ESRF is the first site located outside CXrL where the UNIX SHADOW version has been installed. Although the new SHADOW looks very similar to the old VMS version, it is necessary to consider several differences for running SHADOW correctly.

2 Setting and running SHADOW. Differences with VMS version

SHADOW is installed in different ESRF's Sun machines. It's fully operational in CIVA, FOX and EXPGC. The necessary files are sitting in the

```
~shadow/shadow92
```

directory. In order to run SHADOW from your account you may execute the command

```
source ~srio/.shidl
```

This command executes a script that initializes the SHADOW commands among IDL (see

later) and other useful software. It is a good idea to include this command in your *.cshrc* file. You can also copy and edit this file in order to remove the unnecessary definitions.
Proceed to call SHADOW by *go* command, as always. We recommend use of the prompt mode for the first definition of your source or optical system and then batch mode and menu modes for successive runs. The *start.xx* files contain the namelist with the variable values difined by the user. They can also be directly edited. The file *start*.00 contains the source parameters, *start*.01 the first optical element's parameters, *start*.02 the second optical element, and so on. Appendix A contains the list of all parameters and brief description of their meaning. This help is also available on `~shadow/shadow92/doc` files.
It is also possible to import the *start* files from your VMS session, by copying them with dni or ftp software before running the system on the UNIX machine. This procedure enables the user the continuation in UNIX of the calculations already began in VMS. When importing files from VMS take note of an important difference: *start* files must be split into two different namelists, because SunOS does not accept the total length. This splitting can be done automatically by using the $vax-fix$ utility. The following example is a typical generation of a SHADOW source.

```
dnicp 'diva::disk:[directory]start.01' .
vax-fix start.01
Fixing namelist file start.01 ...  Done
go source
SOURCE selected.  Begin procedure.
Mode selected [?  <ret> for HELP ] ? batch
File containing source description [ start.00 ] ?  start.00
Calling gen_source with /civa/users/b/shadow/shadow92/bin/gen_source start.00
Generated 250 rays out of 5000
...
Shadow::  exit
```

Other utilities have been written for easier use of SHADOW. With *shloop* SHADOW can be run many times, specially with poor statistics system, such as high reflection index crystals or very small slits. This program changes the source seed and runs SHADOW as many times as needed, while renaming the output files.
In the `~shadow/shadow92/demo` directories you can find some examples of running SHADOW for different usual systems, perhaps applied for the ESRF, also including the recent developments in surface errors (waviness and roughness) and crystals in transmission mode. In each directory there is a $*.demo$ file which contains the log of the run, and also can be found the files generated in the demostration.

3 Post-Processes. Results analysis

SHADOW creates *begin.dat*,*star.xx*,*screen.xx*, etc files as in the VMS version. Notice that these binary files are not interchangeable with the VMS computers. In order to analyze these files for extracting needed information and plots we use the basic utilities, $INTENS$, $PLOTXY$ and $HISTO1$. $INTENS$ indicates the total intensity of your file. It is also possible to exclude lost rays, and count the total number of good rays. $HISTO1$ creates a two-column ASCII file containing the histogram of a variable.
$PLOTXY$ plots two columns of your file, as in VMS, but now the output is a device-independent file containing the graphic, named *VP.out*. This device-independent file can be sent to your graphic terminal. If you are using a X terminal (if you work in a Sun workstation or X com-

patible terminal) plot the output by entering:

```
% x11pen VP.out
```

Click inside the image window for its deletion. For a tektronics compatible terminal (when using a Mac select the tek4105 graphic screen, on VersatermPro) enter

```
% tekpen VP.out
```

To create a PostScript file enter:

```
% pspen VP.out > file.ps
```

where *file.ps* is your PostScript file name. Then proceed by printing this file in a PostScript compatible printer.

One of the advantages of using UNIX systems is the possibility of arrowing the inputs. For example, to run BRAGG utility, you can enter the inputs into a file *bragg.inp*, and then type

```
%bragg < bragg.inp
```

Other utilities work are similar to VMS ones, yet there are still some problems with graphics.

4 Analysis with IDL

The major advantages of UNIX SHADOW in speed are diminished because the graphics utilities are not as performant as in VMS. To eliminate this inconvenience IDL software can be used. IDL (Interactive Data Language, by Research Systems Inc) is a very powerful data analysis and graphics software package for scientific use, already licensed and installed in many ESRF's UNIX machines. It is possible to load the SHADOW output files to IDL and perform any analysis inside IDL. I have developed some IDL procedures for analyzing SHADOW files, with similar appearance to the standard utilities, but faster and more performant. These utilities include three-dimensional graphics, color display and graphic manipulation of results. By now three basic utilities are available that permit the standard analysis of SHADOW files. These utilities are, *infosh*, *histo*1 and *plotxy*. Other utilities permit an advanced use as *plot*3 for 3-dimensional graphics and *ray_prop* for watching the propagation of the beam in vacuum, or *plotfile* which allow to plot directly an ascii file on the screen. Open your IDL session just typing *idl*

Basic analysis is simplified by using the three following procedures:

- **infosh:** writes the energy, intensity, lost rays, and coordinates information of the SHADOW structure. The syntax is

  ```
  IDL> infosh,shadow_data,Keywords
  ```

  where *shadow_data* can either be a binary SHADOW file (write it between single quotas) or a SHADOW-IDL structure. For example

```
IDL> infosh,'begin.dat'
```

or, with the same efect

```
IDL> infosh,a
```

where `a` has been stored as indicated below. We higly recommend to store a file in a structure when you plan to use many times the same file for extensive analysis, or simply for obtaining nice graphics. This process optimazes the execution time, because most of the computing time is employed in reading the file. For storing a file in a structure you can also use the optional variable allowed by many of the SHADOW-IDL procedures, as for example:

```
IDL> infosh,'begin.dat',a, nolost
```

where a is the structure name where the *begin.dat* file is stored and *nolost* is an optional keyword to avoid taking into account the loast rays. (here, while studying a source there is not difference between whether or not considering losses). See IDL user's guide for keyword definition and appendix B for all possible available keywords.

- **histo1:** Performs an histogram of a selected column of a SHADOW structure. Syntax is

  ```
  IDL> histo1,shadow_data,column_number,Keywords
  ```

  See apendix B for the list and usage of the keywords.

  Few examples are:

  `IDL>histo1,'begin.dat',11` (performs an histogram of energy)
  `IDL>histo1,'begin.dat',1,nbins=100` (histogram of SHADOW column 1 (X), with 100 bins)
  `IDL>histo1,'begin.dat',3,/ref,/gauss` (histogram of SHADOW column 3 (Z), considering the reflectivity weighted with A**2, A electric field vector, and overplotting a gaussian fit). Remember that for an IDL keyword `/something` is equivalent to `something=1`.

- **plotxy:** plots two columns of a SHADOW IDL structure. Syntax is

  ```
  IDL> plotxy,shadow_data,column_x,column_y,Keywords
  ```

  where keywords are defined in appendix B.
  For example:

  `IDL> plotxy,'star.01',1,3` (performs a real space image of the source)
  `IDL> plotxy,'star.01',4,6,nbins=50` (performs a phase space diagram, with 50 bins in the histograms)

A SHADOW file can be stored directly in IDL by typying:

```
IDL> s = readsh('begin.dat')
```

This command load these files in an IDL structure called source or other chosen name (see IDL user's guide for definition and usage of structures). It is now possible access to all the data

contained in the source generated by SHADOW. Few examples are:

For information stored in source type `IDL>help,/st,source`

For knowing the number of points, type `IDL>print,source.npoint`

For calculating the extrema of Z coordinate type:

```
IDL> print,min(source.ray(2,*)), max(source.ray(2,*))
```
(remember that although in SHADOW column Z is column number three, in IDL is column 2 because all subscripts begin by 0)

For printing the first 100 values of energy (in cm^{-1})
```
IDL> print,source.ray(10,0:99)
```

For duplicating a SHADOW-IDL structure type:
```
IDL>a_bis = a
```

It is also possible to store different SHADOW files with different structure names, and to compare them. To print graphics enter:

```
IDL> set_plot,'ps'
IDL> plotxy,source,1,3
IDL> device,/close
```

These commands select a PostScript file named *idl.ps* for graphic output, then send the graphic generated by *plotxy* to this file, then close the file. *Idl.ps* file can be sent to any PostScript compatible printer (by the command `lp -dps_g31 idl.ps`, where *ps_g*31 is the printer name). After the plot, re-select the graphic terminal, by entering `IDL>set_plot,'x'` or `IDL>set_plot,'tek'`.
During a SHADOW session you perhaps want to go back to the shell mode without closing the IDL session. For that purpose type $Ctrl-Z$ from the IDL session, sending it to the background, then when you want to recuperate the IDL session simply type fg.

IDL is a very powerful data analysis and graphics tool. Many other utilities, such as three-dimensional intensity graphics, contour plots, different colors for lost rays in different optical elements of the beamline, etc. are already available. IDL also enables non-standard analysis such as polarization degree changes along the beamline, etc. See apendix B for a description of the available utilities, or better, you can use the interactive help of IDL, for getting the instructions of use of *plotxy* or whatever other utility, by entering:

```
IDL> doc_library,'plotxy'
```

5 Future Developments

In addition to the amelioration of SHADOW's performances, new physical models for accurate calculations are being introduced, such as new optical elements, automatic optimization of beamline parameters, etc.
A new version containing new graphical applications, A Motif based menu, new optical devices

as komakov lenses and facet mirrors, and other interesting features will be available soon.

6 Where to find more

6.1 Reference

- SHADOW Primer, F. Cerrina (1987)
- SHADOW User's Guide: source description, CXrL (1989)
- SHADOW User's Guide: Utilities, CXrL (1990)

6.2 Papers

- Ray tracing of recent VUV monochromator designs, F. Cerrina, SPIE 503 (1984) 68
- Ray tracing of x-ray optical systems: source models, F. Cerrina, SPIE, (1989)
- SHADOW: A synchrotron radiation ray tracing program, B. Lai and F. Cerrina, Nucl. Instrum. and Meth. A246 (1986) 337
- SHADOW: New developments, B. Lai, K. Chapman and F. Cerrina, Nucl. Instrum. and Meth. A266 (1988) 544
- Modelling of Undulator sources, K. Chapman, B. Lai and F. Cerrina, Nucl. Instrum. and Meth. A283 (1989) 88
- Recent developments in SHADOW, C. Welnak, P. Anderson, M. Khan, S. Singh and F. Cerrina, Rev. Sci. Instrum. 63(1) (1992) 865
- BENT- software for ray-tracing of X-rays on bent oblique-cut crystals: A comment, M. Sanchez del Rio and F. Cerrina, Nucl. Instrum. and Meth. A301 (1991) 589
- Asymmetrically cut crystals for synchrotron radiation monochromators, M. Sanchez del Rio and F. Cerrina, Rev. Sci. Instrum. 63(1) (1992) 936
- A conceptual model for ray tracing calculations with mosaic crystals, M. Sanchez del Rio, S. Bernstorff, A. Savoia and F. Cerrina, Rev. Sci. Instrum. 63(1) (1992) 932
- Waviness effect in ray-tracing of "real" optical surfaces, M. Sanchez del Rio and A. Marcelli, Nucl. Instrum. and Meth. A319 (1992) 170

A Shadow variables

A.1 Column description

1. X spatial coordinate [user's unit]
2. Y spatial coordinate
3. Z spatial coordinate
4. X' direction or divergence [rads]
5. Y' direction or divergence
6. Z' direction or divergence
7. X component of electromagnetic vector (s-polarization)
8. Y component of electromagnetic vector (s-polarization)
9. Z component of electromagnetic vector (s-polarization)
10. Lost ray flag
11. Wavelenght $[cm^{-1}]$
12. Ray index
13. Optical path length
14. Phase (s-polarization)
15. Phase (p-polarization)
16. X component of electromagnetic vector (p-polarization)
17. Y component of electromagnetic vector (p-polarization)
18. Z component of electromagnetic vector (p-polarization)

A.2 Glossary

Glossary/Abbreviation for namelist:

OPD : Optical Path Difference

CDF : cummulative distribution function. in the case of undulators, this is the SHADOW file generated by the utility make_id.

OE : Optical Element

TGM : Toroidal Grating Monochromator

ERG : Extended Range Grasshopper

MRF : Mirror Reference Frame

FWHM : Full Width Half Maximum

SRF : Source Reference Frame

A.3 Source description

fdistr defines source angle distribution types: Available options are: flat(1),uniform(2), gaussian(3), synchrotron(4), conical(5), exact synchrotron(6).

fgrid defines source modelling type – spatial/momentum space. Options are: RANDOM/RANDOM (0), GRID/GRID (1), GRID/RANDOM (2), RANDOM/GRID (3), ELLIPSE in phase/RANDOM on the ellipse (4), ELLIPSE in phase/GRID on the ellipse (5).

fsour spatial source type/shape in X-Z plane. Options are: point (0), rectangle (1), ellipse (2), gaussian (3).

fsource_depth defines the source depth (Y). Options are: no depth (1), flat depth (2), gaussian (3), synchrotron depth (4).

f_coher if generating the A vectors, defines whether the light is incoherent (0), or coherent (1).

f_color photon energy distribution type. Options are: single energy (1), mutliple discrete energies, up to 10 energies (2), uniform energy distribution (3).

f_phot defines whether the photon energy will be specified in eV (0) or Angstroms (1).

f_pol for synchrotron and wiggler sources defines the polarization component of interest: parallel (1), perpendicular (2), total (3).

f_polar flag defining whether or not to generate the A vectors: yes (1), no (0).

f_opd flag defining whether or not to save the optical paths (OPD): yes (1), no (0).

f_wiggler source type. Options: regular/bending magnet/synchrotron (0), wiggler (1), undulator (2).

f_bound_sour flag defining whether or not to optimize the source: yes (1), no (0).

f_sr_type for synchrotron sources, distribution in terms of: photons (0), or power distribution (1).

istar1 seed for random number generator, odd.

npoint number of random rays (0-5000).

ncol source generation routines will fill in the number of columns in your source.

n_circle for fgrid=1,3 and fdistr=5; number of grid points along each circle.

n_color for f_color=2, number of discrete lines in energy, maximum = 10.

n_cone for fgrid=1,3 and fdistr=5; number of grid points along the radius(number of concentric circles).

ido_vx for fgrid=1,3; number of grid points in horizontal angle distribution.

ido_vz for fgrid=1,3; number of grid points in vertical angle distribution.

ido_x_s for fgrid=1,2; fsour=1 points along x, fsour=2 number of concentric ellipses.

ido_y_s for fgrid=1,2; number of points along depth (Y).

ido_z_s for fgrid=1,2; fsour=1 points along z, fsour=2 number of concentric ellipses.

ido_xl for fgrid=4,5; number of sigma levels in X for phase space ellipse source.

ido_xn for fgrid=5; number of rays/sigma level in X

ido_zl for fgrid=4,5; number of sigma levels in X for phase space ellipse source.

ido_zn for fgrid=5; number of rays/sigma level in Z.

sigxl1 for fgrid=4,5; values for sigma levels in X.

sigxl2,....,sigxl10

sigzl1 for fgrid=4,5; values for sigma levels in Z.

sigzl2,....,sigz10

conv_fact for fwiggler=1; conversion factor from meters to user units.

cone_max for fdistr=5; maximum half divergence.

cone_min for fdistr=5; minimum half divergence.

epsi_dx for fdistr=4,6 or fwiggler=1; in X, the distance from the waist that the emittance value corresponds to, signed.

epsi_dz for fdistr=4,6 or fwiggler=1; in Z, the distance from the waist that the emittance value corresponds to, signed.

epsi_x for fdistr=4,6 or fwiggler=1; the beam emittance in units of radians*length units used so far. (in X)

epsi_z for fdistr=4,6 or fwiggler=1; the beam emittance (in Z) in units of radians*length units used so far.

hdiv1 horizontal divergence in +X (radians).

hdiv2 horizontal divergence in -X (radians).

ph1 photon energy: f_color=1 desired energy, f_color=2 first energy line, f_color=3 minimum energy.

ph2 photon energy: f_color=2 second energy line, f_color=3 maximum energy.

ph3 photon energy: f_color=2 third energy line.

ph4,...,ph10

bener for fdistr=4,6 or f_wiggler=1; Electron beam energy (GeV).

pol_angle for f_polar=1; phase difference in degrees.

pol_deg for f_polar=1; degree of polarization (between 0 and 1).

r_aladdin for fdistr=4,6 or f_wiggler=1; bending magnet radius in units of length used for source size, CCW rings negative.

r_magnet for fdistr=4,6 or f_wiggler=1; bending magnet radius (m).

sigdix for fdistr=3; sigma (radians)for horizontal divergence (gaussian angle distribution).

sigdiz for fdistr=3; sigma (radians) for vertical divergence (gaussian angle distribution).

sigmax for fsour=3; sigma in X

sigmay for fsource_depth=3; sigma in Y

sigmaz for fsour=3; sigma in Z

vdiv1 vertical divergence in +Z (radians).

vdiv2 vertical divergence in -Z (radians).

wxsou for fsour=1,2; source width (X).

wysou for fsource_depth=2; source depth (Y).

wzsou for fsour=1,2; source height (Z).

file_traj for fwiggler=1,2; filename containing the electron trajectoryfor wigglers or the CDF's for undulators. (from make_id).

file_source filled in by source generation.

file_bound for f_bound_sour=1; file containing the output of reflag and histo3.

oe_number the last variables are set in their source generation and recorded in END.00 file. They should be left alone

idummy

dummy

f_new

A.4 Optical Element description

fmirr mirror surface: spherical (1), elliptical (2), toroidal (3), parabolical (4), plane (5), codling slit (6), hyperbolical (7), cone (8), polynomial (9).

f_torus for fmirr=3; mirror pole location: lower/outer (concave/concave) (0), lower/inner (concave/convex) (1), upper/inner (convex/concave) (2), upper/outer (convex/convex) (3).

fcyl flag for cylindrical mirror (1), regular mirror (0).

f_ext flag for internal/calculated (0) parameters vs. external/user defined parameters (1). For fmirr=8, f_ext=1.

fstat flag to move the source; yes(1), no(0).

f_screen flag: any screens? yes(1), no (0).

f_plate flag: if n_plates >= 1; image plates orthogonal to optical axis. yes(1), no(0). All plates will have the same orientation.

fslit flag: slit at continuation plane – yes(1), no(0).

fwrite files to save option: all files (0), mirror file only – mirr (1), image file only – star (2), none (3).

f_ripple flag: distorted surface - yes(1), no(0).

f_move flag: move mirror - yes (1), no (0).

f_thick for multilayers: does the thickness of the multilayer vary as the cosine of the angle from the pole - yes (1), no (0).

f_bragg_a flag: is the crystal asymmetric - yes (1), no (0).

f_g_s for f_ripple=1 - type of distortion: sinusoidal (0), gaussian (1), external spline (2).

f_r_ran empty - read in from gaussian ripple file in OE generation.

f_grating flag: is it a grating - yes (1), no (0), if fzp=1, f_grating=1.

f_mosaic if f_crystal=1; flag: mosaic crystal - yes (1), no (0).

f_johansson if f_crystal=1; flag: johansson geometry - yes (1), no (0).

f_side for fmirr=4: focus location at image (0) or source (1).

f_central flag: autotuning of grating or crystal - yes (1), no (0).

f_convex flag: mirror surface concave (0) or convex (1).

f_reflec reflectivity of surface: no reflectivity dependence (0), full polarization dependence (1), no polarization dependence / scalar (2).

f_rul_abs for f_ruling=5: is the line density absolute (0) or signed (1) from the origin.

f_ruling for f_grating=1 - ruling type: constant on X-Y plane (0), constant on mirror surface (1), holographic (2), fan type (3), reserved (4), polynomial line density (5).

f_pw for f_ruling=2 - recording source types: both spherical (0), source plane wave, image spherical (1), source spherical, image plane wave (2), both plane (3).

f_pw_c for f_ruling=2 - recording sources cylindrical: none cylindrical (0), source cylindrical, image spherical (1), source spherical, image cylindrical (2), both cylindrical (3).

f_virtual for f_ruling=2 - source type real/virtual: entrance slit real, exit slit real (0), entrance slit real, exit slit virtual (1), entrance slit virtual, exit slit real (2), both virtual (3).

fshape for fhit_c=1: mirror shape rectangular (0), full ellipse (1), ellipse with hole (2).

fhit_c flag: mirror dimensions finite: yes (1), no(0).

f_mono f_grating,f_central=1 - monochromator type: TGM/ Seya (0), ERG (1), Constant Incidence Angle (2), Constant diffraction angle (3), Hunter (4).

f_refrac flag: reflector (0) or refractor (1).

f_default flag: focii coincident with continuation plane - yes (1), no (0).

f_refl for f_reflec=1,2 - source of optical constants: file generated by PREREFL (0), keyboard (1), multilayers (2).

f_hunt for f_mono=4: first (1) or second (2) grating.

f_crystal flag: crystal – yes (1), no (0).

f_phot_cent for f_central=1: tune to eV(0) or Angstroms (1).

npoint filled in by OE generation.

ncol filled in by OE generation.

scr_number

n_screen for f_screen=1: number of screens.

i_screen for each screen, is it before (1) or after (0) the mirror.

i_abs for each screen, include absorption yes(1), no (0).

i_stop for each screen with stop, obstruction (0), or aperture (1).

i_slit for each screen, is there an aperture stop yes(1), no (0).

k_slit for each stop/slit, specify a shape: rectangular (0), ellipse (1), external (2). (at this point the external shape is not yet implemented.)

istar1

cil_ang if fcyl=1: angle (degrees) of cylinder axis from X axis.

ell_the for f_ext=1, fmirr=2: CCW angle between semi-major axis and mirror pole. for f_ext=1, fmirr=7: angle between major axis and the vector from the hyperbola center to the mirror pole.

n_plates number of extra image plates

ig_seed read in from gaussian ripple file in OE generation.

alpha mirror orientation angle (degrees).

ssour for f_default=0: distance from object focus to the mirror pole.

theta for f_default=0: incidence angle (degrees).

simag for f_default=0: distance from mirror pole to image focus.

rdsour for fstat=1: source distance from mirror pole.

rtheta for fstat=1: incidence angle (degrees).

off_soux for fstat=1: offset in X in Mirror Reference Frame (MRF).

off_souy for fstat=1: offset in Y in MRF.

off_souz for fstat=1: offset in Z in MRF.

alpha_s for fstat=1: source rotation around Z (CCW +).

rlen1 fshape=1: mirror half length +Y. fshape=3: internal minor axis (Y).

rlen2 fshape=1: mirror half length -Y. fshape=2,3: external outline minor axis (Y).

rmirr f_ext=1, fmirr=1: spherical mirror radius.

axmaj f_ext=1, fmirr=2,7: semi-major axis

axmin f_ext=1, fmirr=2,7: semi-minor axis

cone_a f_ext=1, fmirr=8: cone half opening (degrees).

r_maj f_ext=1, fmirr=3: tangential (major) axis.

r_min f_ext=1, fmirr=3: sagittal (minor) axis.

rwidx1 fshape=1: mirror half width +X. fshape=3: internal major axis (X).

rwidx2 fshape=1: mirror half width -X. fshape=2,3: external outline major axis (X).

param f_ext=1, fmirr=4: parabola parameter P of Y**2 = 2 P X.

hunt_h f_mono=4: monochromator height (distance between beams.)

hunt_l f_mono=4: monochromator length.

blaze f_mono=4: blaze length (degrees).

ruling f_ruling=0,1: ruling density in l/cm. f_ruling=3: ruling density at center l/cm. f_ruling=5: ruling density at the origin, i.e. a0 of polynomial.

order diffraction order, negative inside (European convention). It is also used in Laue Crystals to indicate incidence onto the bragg planes (order=+1) or incidence below the bragg planes (any other value for order).

phot_cent for f_phot_cent=1: photon energy (eV) to autotune grating/crystal to.

x_rot f_move=1: rotation about X axis (degrees, +CW).

d_spacing calculated.

a_bragg f_bragg_a=1: angle between crystal planes and surface.

spread_mos f_mosaic=1: mosaic spread FWHM (degrees).

thickness f_mosaic=1: crystal thickness cm.

r_johansson f_ext=1: johansson radius.

y_rot f_move=1: rotation about Y axis (degrees, +CW).

z_rot f_move=1: rotation about Z axis (degrees, +CW).

offx f_move=1: mirror offset in X (MRF).

offy f_move=1: mirror offset in Y (MRF).

offz f_move=1: mirror offset in Z (MRF).

sllen fslit=1: slit length (sagittal, X).

slwid fslit=1: slit width (dispersion, Z).

sltilt fslit=1: slit tilt (+CCW, degrees).

cod_len fmirr=6: codling slit length (sagittal).

cod_wid fmirr=6: codling slit width (effective, tangential).

x_sour fstat=1: source offset in X (source reference frame -SRF).

y_sour fstat=1: source offset in Y (SRF).

z_sour fstat=1: source offset in Z (SRF).

x_sour_rot fstat=1: source rotation about X (SRF).

y_sour_rot fstat=1: source rotation about Y (SRF).

z_sour_rot fstat=1: source rotation about Z (SRF).

r_lambda f_phot_cent=1: Angstroms to autotune grating/crystal to.

theta_i f_plate=1: elevation (degrees) Y=0 plane rotated about the X axis, +CCW.

alpha_i f_plate=1: azimuth (degrees) Y=0 plane rotated about the Z axis, +CCW.

t_incidence incidence angle (degrees), defining optical axis.

t_source source plane distance

t_image image plane distance

t_reflection reflection angle (degrees).

file_source for first OE, filename of SOURCE array. if unchanged, BEGIN.DAT.

file_rip for f_ripple=1, f_g_s=2,3: file containing the distortion information.

file_refl for f_crystal=1: file containing the crystal parameters. for f_reflec=1,2 and f_refl=0: file with optical constants. for f_reflec=1,2 and f_refl=2: file with thicknesses and refractive indices for multilayers.

file_abs for each screen with absorption, file containing optical constants.

file_mir for fmirr=9(polynomial): file containing the polynomial description.

thick for each screen including absorption, thickness of the film in cm.

rx_slit for each slit/stop in a screen, dimension in X.

d_plate mirror center to the plate.

rz_slit for each slit/stop in a screen, dimension in Z.

sl_dis for each screen, absolute distance from the mirror.

fzp flag: Fresnel Zone Plate – yes(1), no(0). set f_grating=1.

holo_r1 f_ruling=2: defining the creation of the hologram: entrance slit distance (cm).

holo_r2 f_ruling=2: exit slit distance (cm).

holo_del f_ruling=2: entrance slit incidence angle (degrees).

holo_gam f_ruling=2: exit slit incidence angle (degrees).

holo_w f_ruling=2: recording wavelength (angstoms).

holo_rt1 f_ruling=2: entrance slit rotation angle (degrees).

holo_rt2 f_ruling=2: exit slit rotation angle (degrees).

azim_fan f_ruling=3: position of ruling focus on mirror plane. angle from Y, degrees, +CCW.

dist_fan f_ruling=3: distance from grating center (cm).

coma_fac f_ruling=3: coma correction factor, 0 if none.

alfa f_refl=1: alpha of the complex dielectric constant defined by: epsilon = (1-alpha)+i gamma

gamma f_refl=1: gamma of the above equation.

r_ind_obj f_refrac=1: index of refraction in object space.

r_ind_ima f_refrac=1: index of refraction in image space.

rul_a1 f_ruling=5: a1 coefficient of polynomial.

rul_a2 f_ruling=5: a2 coefficient of polynomial.

rul_a3 f_ruling=5: a3 coefficient of polynomial.

rul_a4 f_ruling=5: a4 coefficient of polynomial.

x_rip_amp f_g_s=0: amplitude of distortion along X axis.

x_rip_wav f_g_s=0: wavelength along X axis.

x_phase f_g_s=0: phase for X axis; 0 implies maximum at origin.

y_rip_amp f_g_s=0: amplitutde along Y axis.

y_rip_wav f_g_s=0: wavelength along Y.

y_phase f_g_s=0: phase for Y axis.

n_rip 0 - read in from gaussian ripple file in OE generation.

f_roughness surface roughness flag

file_rough when f_roughness=1, name of the file with the 2-dimensional power spectral density (PSD) function.

rough_x the roughness rms value along the X axis

rough_y the toughness rms value along the Y axis

oe_number the rest of the variables are filled in during OE generation.

idummy

dummy

B IDL utilities

B.1 Basic utilities

- **procedure INFOSH** Information about the selected SHADOW structure in IDL

  Usage: `IDL> infosh, shadow_in, Keyword_parameters`
  where `shadow_in` could either be a SHADOW file written between single quotes (i.e. `'begin.dat'`) or a SHADOW structure stored in IDL.

  Keyword parameter :

  NOLOST = consider losses
  0 all rays (default option)
  1 exclude losses

  POLARIZATION = when set to 1 calculates the polarization parameters:
  I_s (total intensity for s-polarization)
  I_p (total intensity for p-polarization)
  S_0 Stokes = $|E_s|^2 + |E_p|^2$
  S_1 Stokes = $|E_s|^2 - |E_p|^2$
  S_2 Stokes = $2|E_s||E_p|cos(phase_s - phase_p)$
  S_3 Stokes = $2|E_s||E_p|sin(phase_s - phase_p)$

- **procedure HISTO1** an IDL implementation of the HISTO1 SHADOW's utility

  Usage: `IDL> histo1, shadow_in, column, shadow_out, Keyword_parameters`
  where `shadow_in` is a SHADOW file or structure with the data and `shadow_out` (optional parameter) is the name of the IDL-SHADOW structure where store the file.

  Parameters :
  shadow_in is a SHADOW file or structure with the data
  column number of column to analyze
  shadow_out (optional parameter) is the name of the IDL-SHADOW structure where store the file.

  Keyword parameters :

  XRANGE = [min,max], limits of the histogram.
  YRANGE = [min,max], Height limits of the histogram.
  NBINS = number of bins (default = 25)
  NOLOST = consider losses
  0 all rays (default option)
  1 exclude losses
  REF = include reflectivity
  0 no (default)
  1 yes (A^2 as weight= column 23)
  n ($21 < n < 34$), n=column to weight (see PLOTXY for column description)
  GAUSSFIT = makes a gaussian fit of the histogram
  0 No (default)
  1 Yes
  TITLE = top title
  XTITLE = title in X column

YTITLE = title in Y column

OVER = overplots the previous histogram when set to 1.

WRITE = when set to 1 writes the histogram in the file HISTO1.

- **procedure PLOTXY** an IDL implementation of the PLOTXY SHADOW's utility

Usage: `IDL> plotxy,shadow_in,col1,col2,shadow_out,Keyword_parameters`

Parameters :

shadow_in may be a SHADOW file (between quotes) or an IDL-SHADOW structure

col1 x column to plot

col2 y column to plot

the column meaning is the following:

1-18 Regular columns (see appendix A.1)

20 $R = sqrt(X^2 + Y^2 + Z^2)$

21 angle fron Y axis

22 the magnitude of the electromagnetic vector

23 $|E|^2$ (total intensity)

24 total intensity for s-polarization

25 total intensity for p-polarization

30 S_0 Stokes = $|E_s|^2 + |E_p|^2$

31 S_1 Stokes = $|E_s|^2 - |E_p|^2$

32 S_2 Stokes = $2|E_s||E_p|cos(phase_s - phase_p)$

33 S_3 Stokes = $2|E_s||E_p|sin(phase_s - phase_p)$

shadow_out (optional parameter), name of the SHADOW-IDL structure where to store the file.

Keyword parameters :

XRANGE = [min_x,max_x]

YRANGE = [min_y,max_y]

NOLOST = consider losses

0 all rays (default option)

1 exclude losses

2 only losses

NBINS = number of bins (default=25)

CART = boolean when if present and non-zero forces the data space to have cartesian coordinates (i.e. square has aspect ratio one)

NOCOMM = comment column

0 with comment section (default)

1 without comment section

HISTO = histogram setting

-1 skip histograms

0 with histograms (default)

1 overplot histogram weighted with intensity

TITLE = chain for title

SLIT = [lb_x,lb_y,ru_x,ru_y] where lb_x and lb_y are the left bottom vertex of the slit and ru_x and ru_y are the coordinates of the right upper vertex of the slit.

GAUSSFIT =makes gaussian fits of the histograms

PUBLISH = flag to supress graphic output not suitable for publication, i.e. username, date, time, filepath, etc. When flag is present and non-zero, output is supressed.

CONTOUR = makes contour curves:
0 (default) scattered plot
1 makes contour curves (no reflectivity is considered)
2 makes contour curves including reflectivity.

COLOUR KEYWORD PARAMETERS

CCOL = Column by which to color code scatter plot points. If this parameter is present and non-zero, each point will be plotted with a color which depends on the value in the column specified and the current color table. If the parameter is an integer, it is interpreted as the column number of the data to use for color information. If the parameter is an array, it is interpreted as the actual data by which to color the arrays. One may use the IDL utility XLOADCT to change interactively the color table.

CLIMITS = [min,max], the limits applied to the values in the color information column, the default is to use the range of the data in that column.

CBINS = number of bins for the color histogram (default=100)

CREF = when set to 1 takes into account reflectivity in the color histogram (default=0, no reflectivity)

CFILE = name of the SHADOW file or IDL-SHADOW structure from where the color column (if integer) is read. To be set when the color reference file is different from the one in `shadow_in`

B.2 Advanced utilities

PLOT3 plots a three-dimensional histogram of the intensity from a SHADOW file or structure. See description of use and available keywords in `IDL> doc_library,'plot3'`

RAY_PROP plots the evolution of an image (columns 1,3) in vacuum along the beamline axis. See description in `IDL> doc_library,'ray_prop'`

B.3 Tools

READSH Load a SHADOW file in a IDL-SHADOW structure. It is also possible to store many files into only one structure. This is specially useful when running SHADOW in a loop. See complete description in `IDL>doc_library,'readsh'`

PLOTFILE Plots a N-column ascii file with IDL. See `doc_library, 'plotfile'`

PSPRINT Help the user to create and send a PostScript file containing the IDL graphic to any ESRF PostScript printer.