Helmholtzzentrum Berlin

Beginners Guide to PHASE

A brief introduction to the physical optics code PHASE

Johannes Bahrdt, Uwe Flechsig, M. Bowler
25.2.2021

Overview and Functionality of the PHASE Package

This document is intended to allow new users to get started using the software package PHASE, primarily by going through simple examples. It will not cover all the functionality of the code, nor the mathematics behind the code. Please, visit the HZB-homepage:

https://www.helmholtz-berlin.de/forschung/oe/wi/undulators/arbeitsgebiete/phase_en.html

On this page, you find links to the github repository of all sources

https://github.com/flechsig/phase

and of installation instructions

https://github.com/flechsig/phase/wiki

For a description of the mathematics behind the code we refer to the reference as listed at the end of the document.

The co-ordinate system is shown in Figure 1. The x-direction is along the beamline, y is vertical and z horizontal. (x,y,z) defines the source position of a ray before an optic, (x',y',z') defines the image point of that ray. (w,l) gives the intersection of the ray on the optic, and u(w,l) gives the profile of the optical surface. The optical surface is characterized by an 8th order polynomial.

Figure 2 shows schematically the layout of a beamline with four elements. Each element (m) is enclosed by two planes Im and Im+1. The transformation between two succeeding planes is described by a matrix. The transformation over the complete beamline is given by the product of the individual matrices.

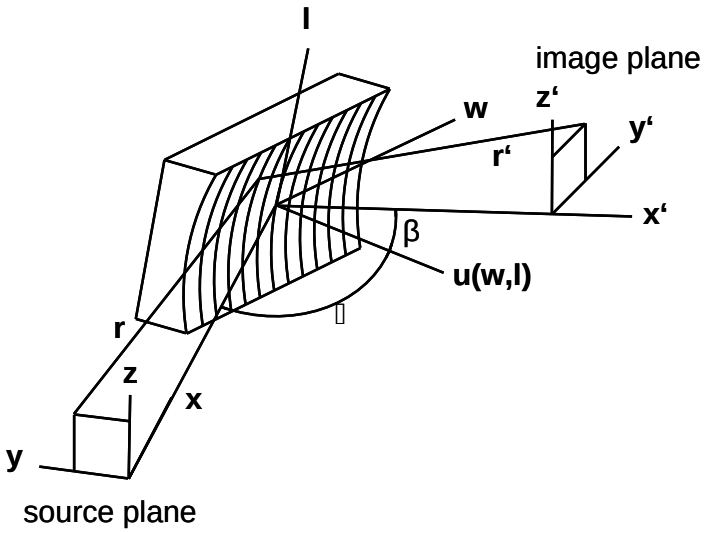


Figure 1: Co-ordinate system for an optical element.

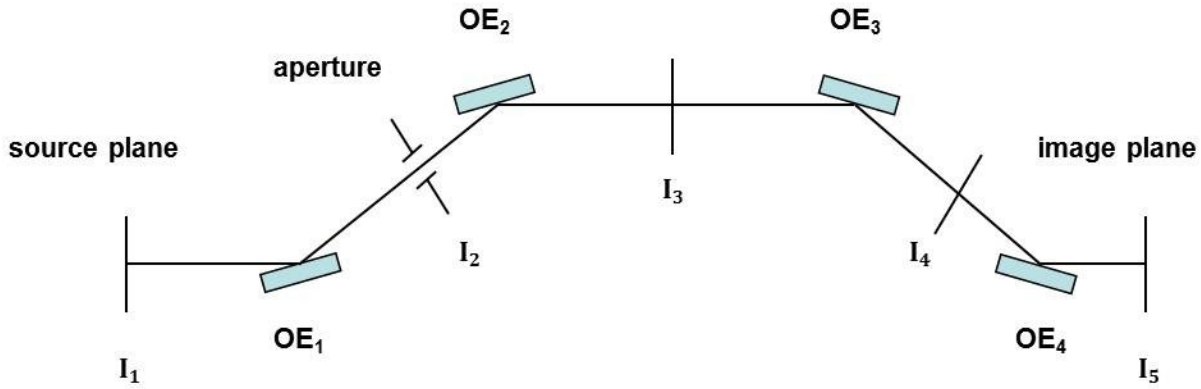


Figure 2: Schematic layout of a beamline with four optical elements (OEs) and five interfaces (I_1 – I_5).

The software package PHASE consists of several parts:

- *phaseqt* (*qt* GUI): *phaseqt* can be used for both geometric ray-tracing and propagation of coherent light along a beamline. It is recommended to use ray tracing part to check the beamline description first, before attempting to carry out the more time consuming wavefront propagation. The modern Qt environment provides easy means for parallelization.
- *phase* (Motif GUI): The numerics of the Motif and the Qt version are identical because both are based on the same set of mathematical routines. However, the Motif interface is not maintained anymore due to increasing problems with compatibility of Motif libraries.
- *phasesrv*: This PHASE version has the same mathematics and IO-files as the other versions, however, it needs only a minimum number of libraries, e.g. there is no GUI and this code has to be started in batch mode. The slim architecture guarantees a high degree of compatibility with many computer systems.
- *phaseopti* is a package for the optimization of beamline parameters in geometric optics mode. The CERN routine MINUIT is utilized for the multi-parameter optimization. The GUI of *phaseopti* is still Motif based. The input file for *phaseopti* defining the beamline parameters is generated from *phaseqt* (or *phase*).
- *phaseextract* is tools for the extraction of transformation maps of single optical elements or a complete beamline,
- (PHASE4IDL, Obsolete, has been replaced with phython scripts, see github repository)
- PHASEphython is a tool box with several Fourier Optics propagators. These propagators are well suited for free drift propagation. In addition various routines are provided for electric field generation (Gaussian beams) and field array manipulation. These routines are written in C language and they can be called via specific *.pros from IDL. These routines are described in detail at PHASE wiki. phaseqt can be started in batch mode from this interface, as well.
- PHASE live CD: a live CD image can be downloaded from the PHASE homepage. http://www.helmholtz-berlin.de/forschung/grossgeraete/undulatoren/arbeitsgebiete/phase_en.html
  This bootable CD can be used with any windows or linux system. The image contains a full linux system, a complete PHASE installation and all libraries required to run all individual parts of PHASE.

(PHASE INSTALLATION, OBSOLETE, FOLLOW INSTRUCTIONS AT GITHUB)

A PHASE_HOME environment variable must be defined, which defines the directory main PHASE directory. Binaries and libraries are located in $PHASE_HOME/bin and $PHASE_HOME/lib. IDL routines are in $PHASE_HOME/idl and $PHASE_HOME/share contains several text files.

The location of the libraries has to be defined (this is just an example):

export $PHASE_HOME=$HOME/xxx

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/intel/composerxe-2011.0.084/compiler/lib/intel64
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/net/group/idev/common/usr_dinux5/usr/local/qwt-6.0.1/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/net/group/idev/common/usr_dinux5/usr/local/hdf5-1.8.6/lib.

Simulations should be done in a specific working directory. PHASE creates several output files in the working directory, e.g. files describing the beamline, the individual optical elements, output rays and fields.

Correct operation of the code also requires the LANG environment variable to be set to c, using the command in Linux

export LANG=c

Graphical User Interface

Overview

PHASE is started from a linux window with the command *phaseqt*. The main window (Figure 1) shows up and the commands and further information and error messages are printed to the linux screen. The main window is arranged in the following manner:

- A bunch of pull down menus on the upper left is used to define the sources, initiate the simulations and write the results to files.
- The left column on the main window is used to arrange the Optical Elements in a beamline, and to define a series of parameters required for the simulations
- In the center column the results are plotted and some additional information are given, such as center of gravity and rms /FWHM width /height.
- At the top of the right column the source is defined. The rest of the right column is used to define the OE which has been selected in the beamline box. Three sets of parameters are required for each OE:
    - Parameter defining the surface shape of the OE (e.g. toroidal, elliptical, conical) and line density parameters in case of a grating
    - Position and orientation of the OE in the beamline
    - Errors of the OE surface shape (slope errors), positioning errors (misalignment), finite size of the OE which may cut the beam

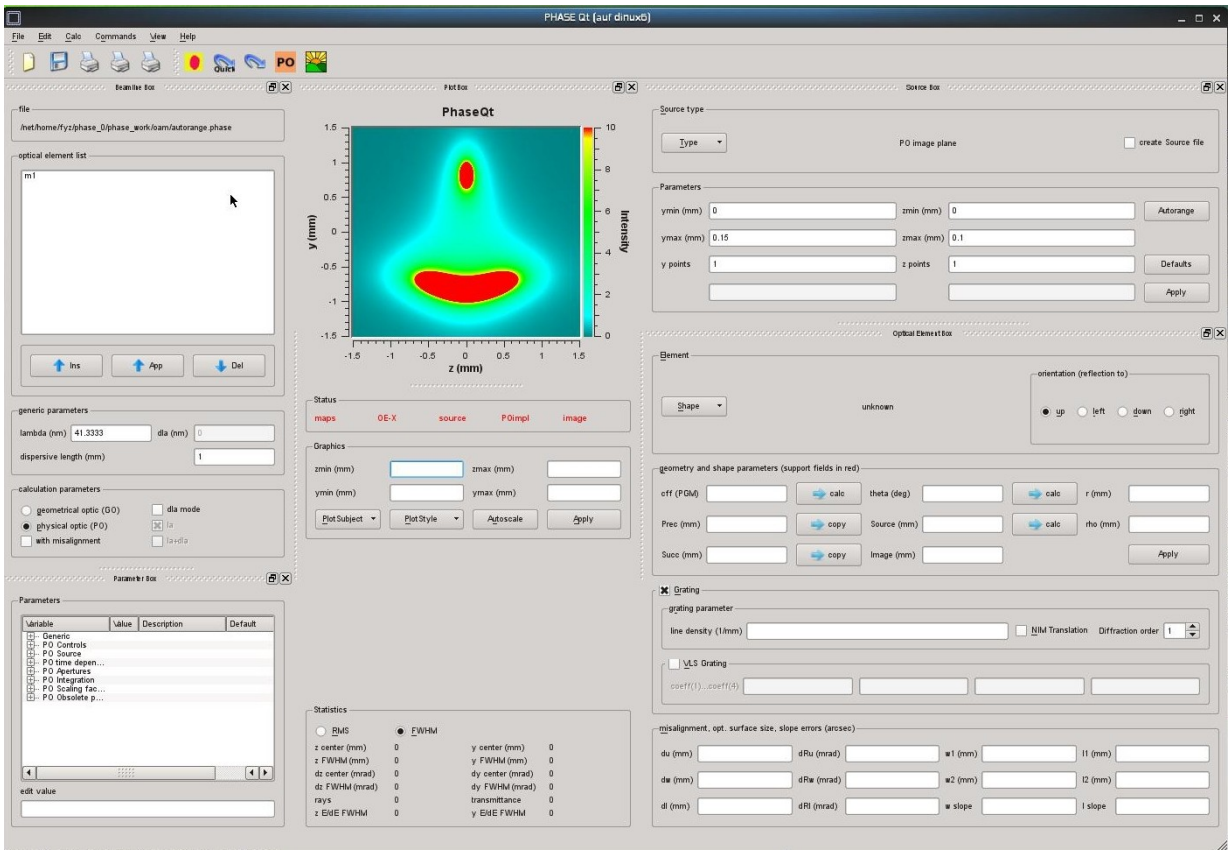
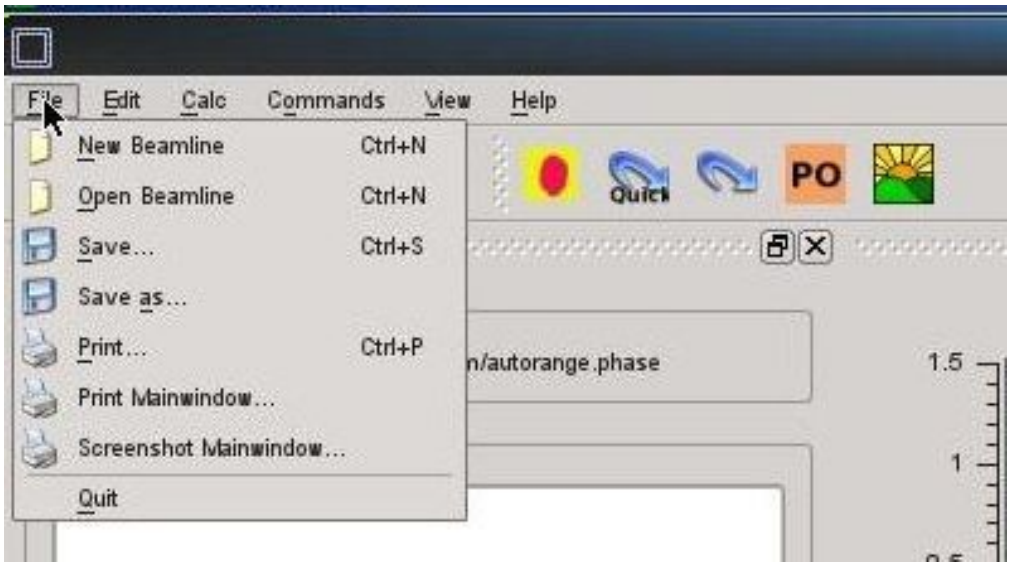


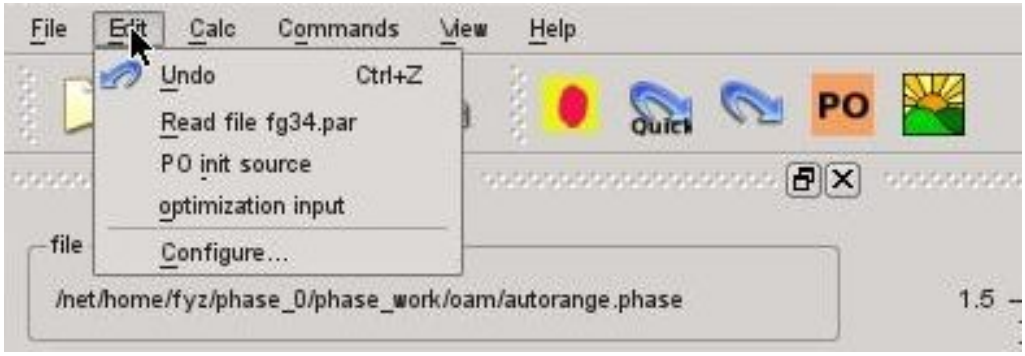
Figure 1: Main menu for *phaseqt*

File Menu



The complete information of a beamline with the exception of the source data is defined in the s-called beamline file. Existing or new files can be opened and saved from this menu.

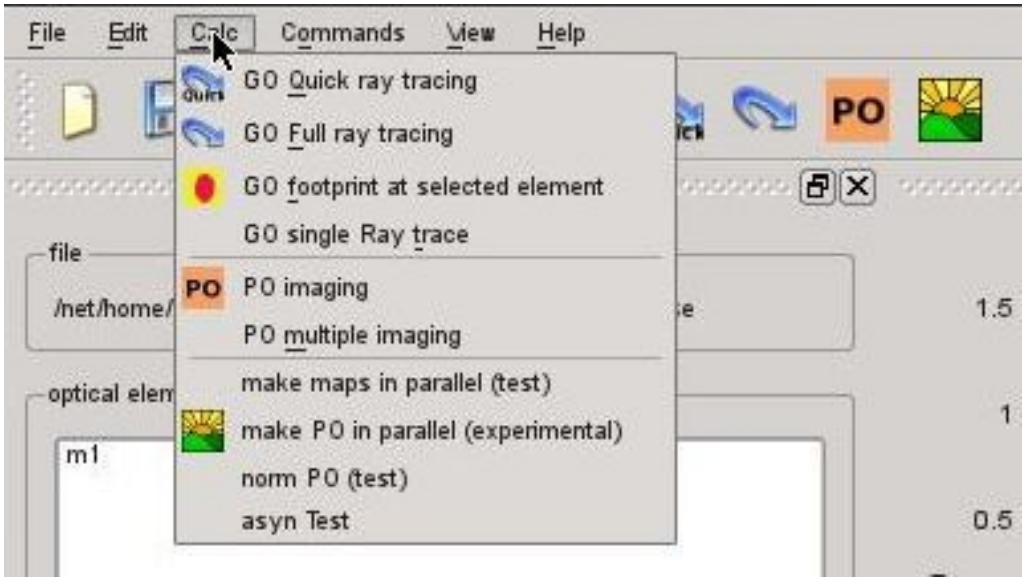
Edit Menu



The edit menu is used to define and initialize sources for the PO mode and to define the optimization input. The source definition is done under Configure. Depending upon the source type, WAVE files (source4) or hdf5 files (source can be defined. Source6 (brightness) is not implemented yet.

The parameter file fg34.par is obsolete since all parameters are defined in the beamline file.

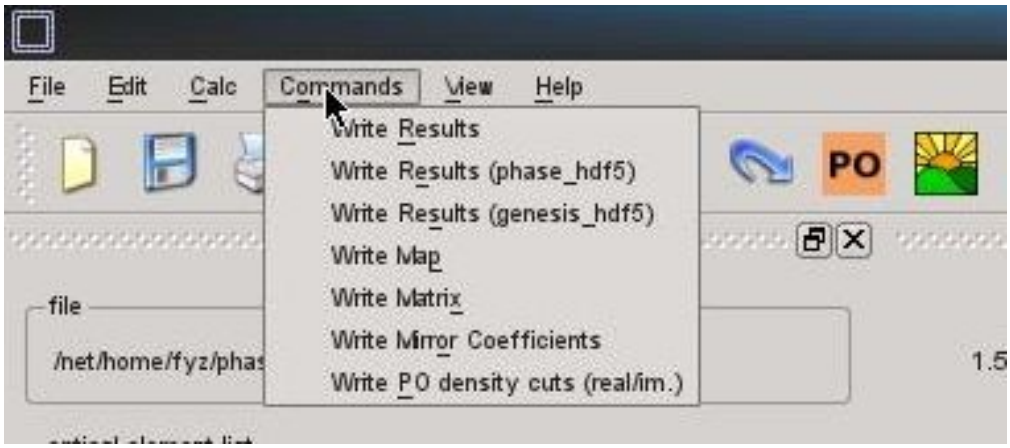
Calculation Menu



This menu defines the operation to be carried out. Most of these operations are directly accessible via a symbol from the main panel. The operations are:

- quick raytracing excluding slope errors and beam cutting by finite OE size
- full raytracing including slope errors and beam cutting by finite OE size
- footprint on OE in raytracing mode
- single frequency Physical Optics (PO) propagation in SPA mode
- Multiple frequency PO-SPA propagation
- Make maps in multi threading mode
- Single frequency PO-SPA propagation using all available threads
- Norm PO
- Asyn test

Write Files Menu



VIEW

INPUT AND OUTPUT FILES

The following table gives a summary of the input / output files.

Filename	data	reading	writing
test.phase	beamline file	File / open	Files /save
test.inp	input rays (4 dim.)	edit / source / source form file	edit / source / create source file
test.out	output rays		command / write result
m1	mirror coefficients	Push apply button in mirror box	Command / write mirror coefficients (mirror-coefficients.dat)
m1.pcke	optical element (OE) description	starting phase	beamline / element / apply
m1.pckg	geometry of OE in beamline	starting phase	beamline / geometry / apply or beamline / element / apply
m1.omx	transfer matrix (70x70)		???
m1.map	transfer map		commands / write map
m1.isec	transfer map to OE surface		???
test.pcko	parameters for optimisation		edit / optimisation / ok
m1.datg	not used		
m1.date	not used		
test.pck	not used		
test.pcl	not used		
eyre.dat etc	electric fields, file names are defined in fg34.par	edit / init source	command / write result

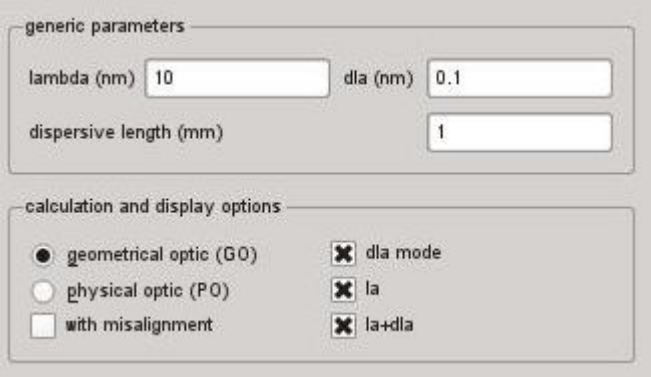
br4d.dat	4 dim. brightness (under development)	edit / init source	
phase.pck	all file names	starting phase	files / ok

RAY TRACING

OVERVIEW

PHASE permits Ray Tracing (RT) within Geometric Optics. Due to the matrix representation of the optical elements the simulations are extremely fast in simple RT mode (without slope errors or OE sizes. Simple RT runs runs are always recommended for setting up the beamline and checking the input data. Only in a 2nd step full RT or PO propagation is done. A rectangular grid of rays for the input, focused by a toroidal mirror is used for the first example. The second example will look at an undulator source, followed by a KB mirror pair.

BASIC PARAMETERS



- Define basic wavelength for setting up the grating geometry (if grating is included)
- Chose incremental wavelength change dla if dla-mode is activated
- The dispersive length is the monochromator exit arm length and the value is used to evaluate the energy resolution of the monochromator.
- Switch to Geometric Optics in the calculation parameter box
- Choose whether a misalignment of the OEs is to be included
- Select dla-mode:
  dla mode = 0: geometry is created and propagation is performed with the wavelength lambda.
  dla_mode = 1: geometry is created with lambda and propagation is done with wavelength lamda and lambda + d-lambda. Note: For a grating the light is centered at the exit slit for dla_mode = 0. For dla_mode = 1 two spots displaced along the dispewrsion plane are generated (Figure xx).

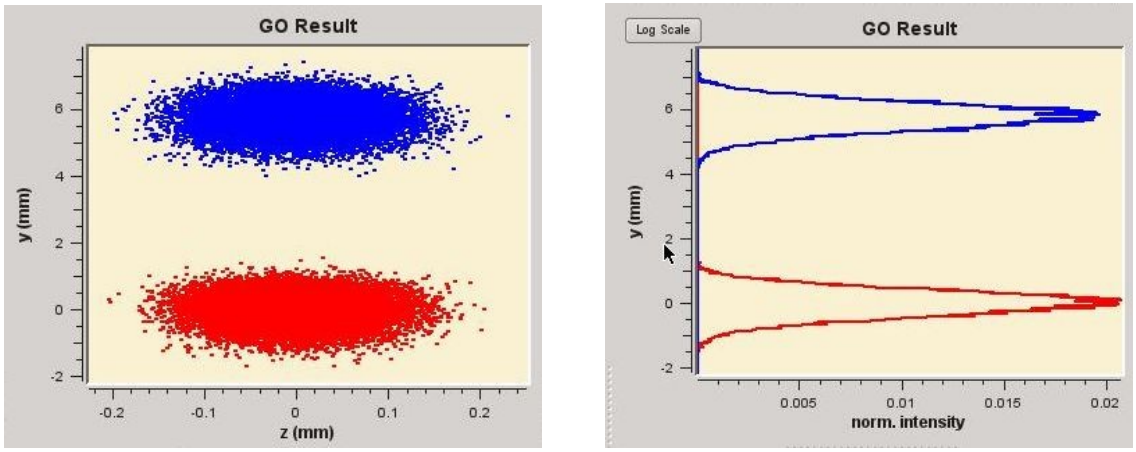
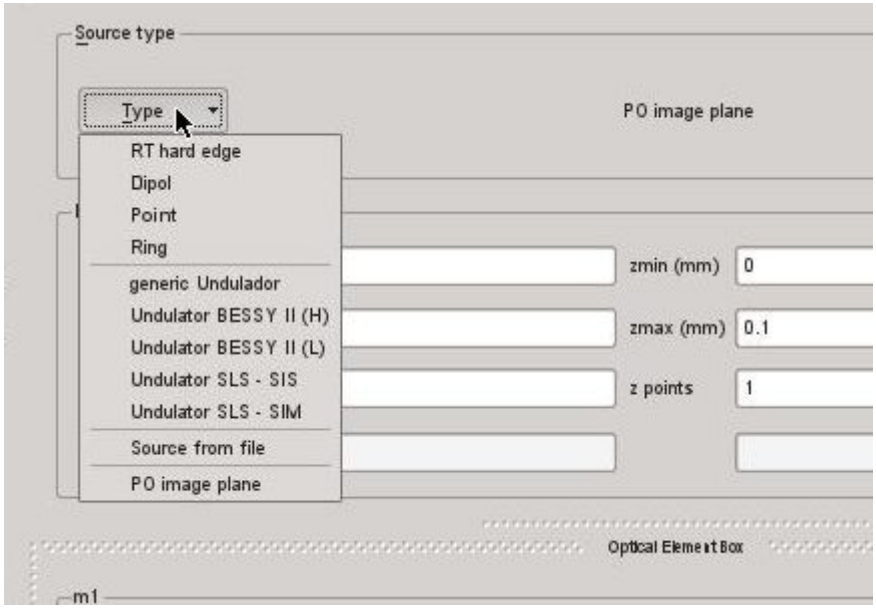


Figure xx: Ray tracing result behind a grating in the exit slit plane. The two wavelength show up at different positions in the dispersion plane if dla_mode = 1.

SOURCE DEFINITION



Most of the sources are dedicated to GO mode. For PO mode the image size and granularity has to be defined under PO image plane. The GO sources are:

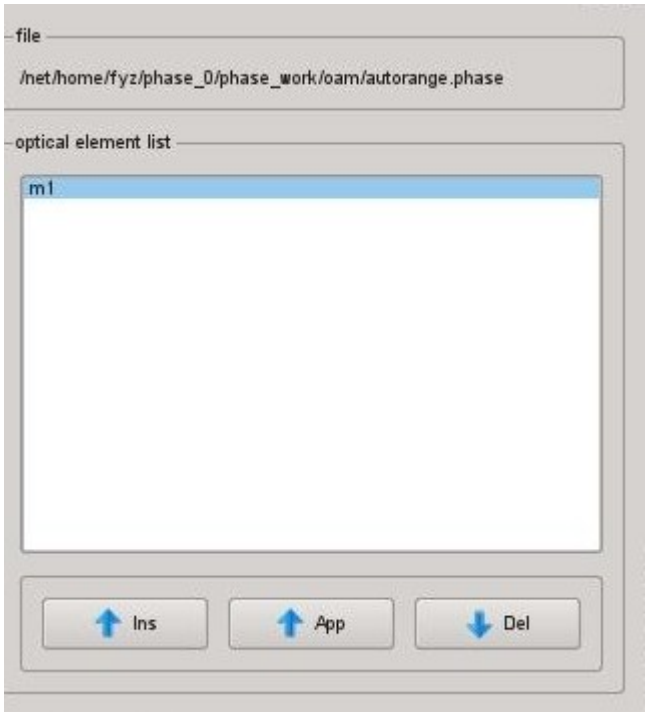
The GO sources which are available are listed in the following table

Menu point	Comment
Hard edge	source size and divergence are rectangular in both planes and the rays are equidistant
Point source	Gauss'schen distribution hor. and vert. source sizes and divergencies are defined by sigma values
Dipole	Gaussian distribution for size and vert. divergence, rectangular for hor. divergence, randomly distributed rays, definition by sigma values eccept of half width for the hor. divergence
Ring	

Single ray	this option is useful for debugging the beamline data
Generic undulator	various undulators with hard wired betatron functions, or a generic undulator with input beam dimensions
BESSY II Undulator (high beta)	
BESSY II Undulator (low beta)	
SLS Undulator SIS	
SLS Undulator SIM	
input from file	Used in geometrical optic (ray tracing). The first line give the number of rays. The rays are defined via four columns which give the vert. and hor. displacement (mm) and the vert. and hor. divergence (rad).
PS image plane	In case of physical optic the source data are read from a file. This mask defines the grid points in the image plane where the intensities have to be calculated.

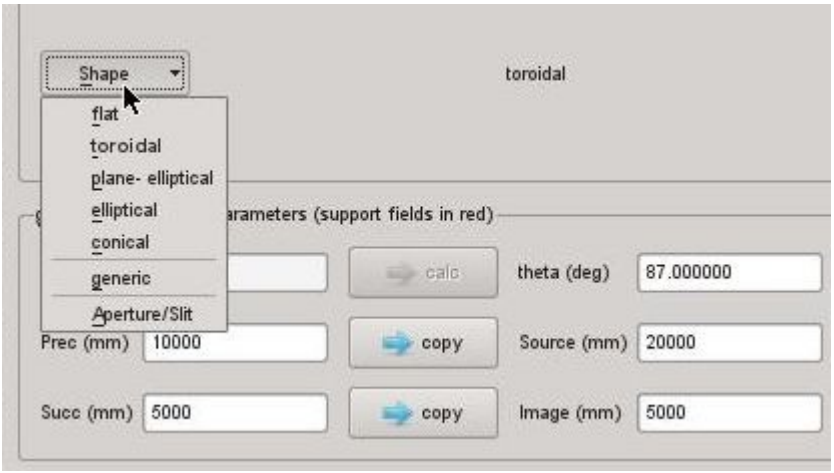
After a source is selected, the beam sizes and divergences are adjusted and the number of rays is defined, the source is created with the apply button. A source file is written if the *create source file* flag is enabled (only RT mode). The *autorange* button needed only for PO mode (see PO Integration).

Beamline Layout



If a new beamline is generated insert elements in the beamline box. After highlighting an Optical element this element can be edited I the OE box. Elements which are not needed can be deleted with the delete button. When pushing the Apply button the beamline is built, i.e. the matrices of the individual OEs are built and multiplied with each other to get one matrix for the complete beamline. Note: Only in simple RT mode the complete beamline can be treated with one matrix. For full RT (including slope errors and OE sizes) the RT is performed step by step.

OPTICAL ELEMENTS



Then select edit / beamline. At the top of the beamline form (Error: Reference source not found) is the name of a beamline file. If this file exists, the information on a pre-defined beamline will be read. Information on the beamline will be written to this file on exiting the window. If starting from scratch, hit the left pointing arrow button to add optics and enter the root of the name for files that will store the details of the first optics, eg m1, in the file selection box. Click on that name in the scroll down box to highlight it, and a form (Error: Reference source not found) allowing the definition of the optical element appears. For ray tracing, the geometrical optic choice should be checked, and note that the wavelength is set in the beamline form. All calculations are carried out at a single wavelength. For a dispersive element (e.g. line grating) the position of the rays in the focal plane can be expressed in equivalent wavelength shifts when filling the dispersive length box. The quantities are related via:

$$\partial\lambda = \partial y \cdot \frac{d}{n} \cdot \cos(\beta) \cdot \frac{1}{L}$$

where d is the line spacing, n is the order and L is the dispersive length (e.g. the distance between a spherical grating and the exit slit for an SGM). Vertical dispersion is assumed. Choosing the plotting option footprint the histogram versus $\partial\lambda$ is plotted in lower right box.

The geometry of the optical element is set in the upper part of the optical element form. The Prec and Succ boxes define the continuation planes which enclose the element, see Error: Reference source not found. The set of rays will be transformed from the preceding to the succeeding planes. These need not correspond to focal planes of the system, the distances to which are given in the source and image boxes in the form. Specify the angle of incidence, and click the 2 arrows which will generate the correct radii for the toroidal mirror. The cff parameter is only needed for PGMs - as a general rule parameters which do not apply to the optic in question are just ignored. If a PGM were being described, clicking the arrow box next to cff will automatically set the required incident angle.

The mirror type is defined in the box element type. Any arbitrary 5^{th} order polynomial can be used to describe the surface. In this case the option coefficient file has to be chosen. The coefficient file name equals the file name in the beamline box without any extension. The surface profile is defined via

$$u = \sum_{ij} a_{ij} \cdot w^i l^j$$

where the factors a_{ij} are given in the coefficient file in the following order:

I	j	a_{ij}
0	0	a_{01}
0	1	a_{02}
0	2	a_{03}
0	3	a_{04}
0	4	a_{05}
0	5	a_{10}
1	0	a_{11}
Etc		

Only the coefficients which are non zero have t o be given. For this example, nothing else is required - hit apply and then OK which will generate the correct transfer matrix.

From the commands menu, select calculation, then ray tracing. To visualise the data, select commands then graphic (Error: Reference source not found). The upper drop down menu in the graphic box defines the type of plot, scatter, surface, contour etc, and the lower what is to be plotted. If you select result, and footprint, click apply, 4 plots will appear in the graphics window, the scatter plot in the image plane, the horizontal and vertical distributions and the wavelength (Figure 2).

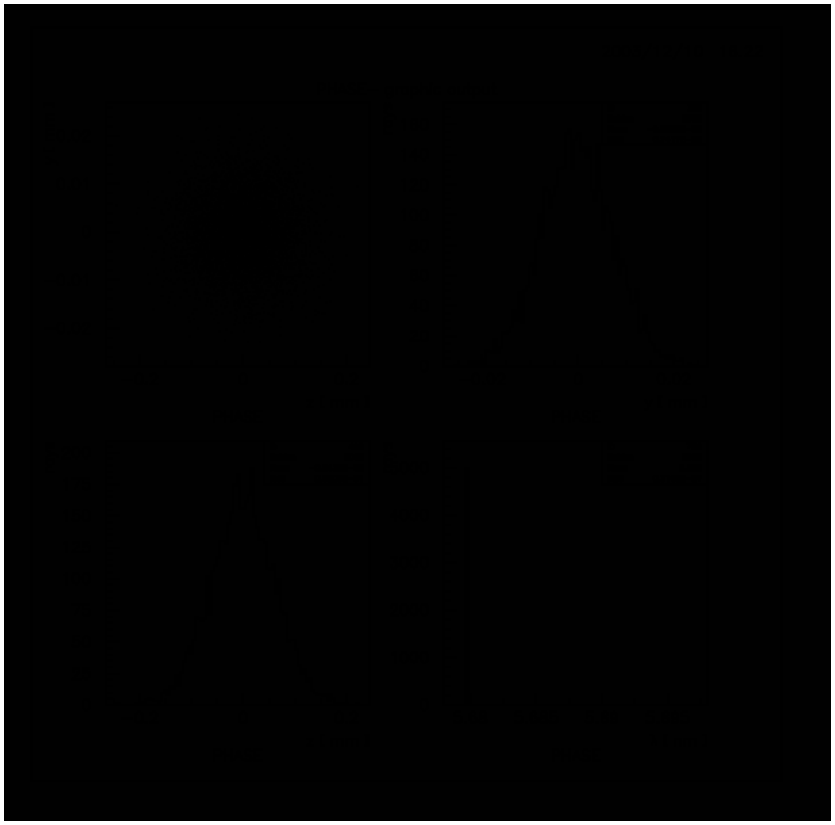
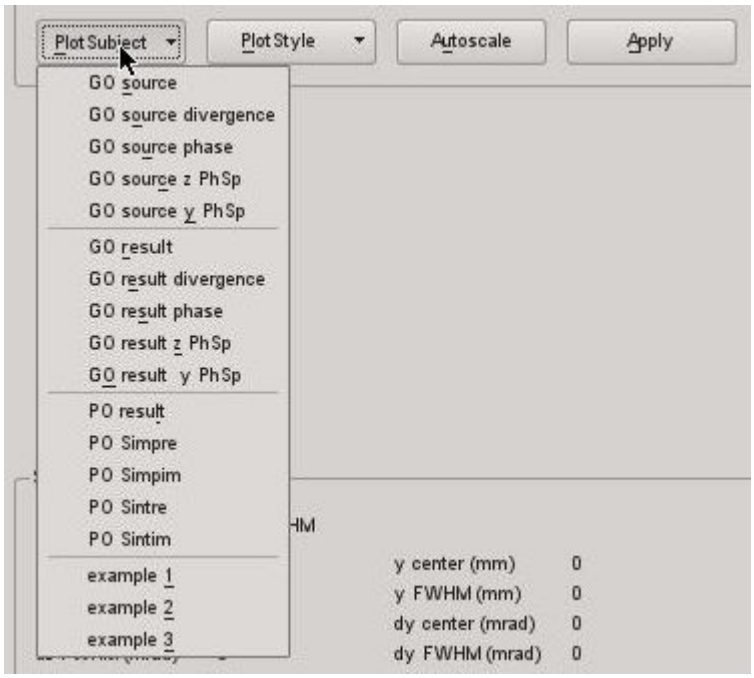
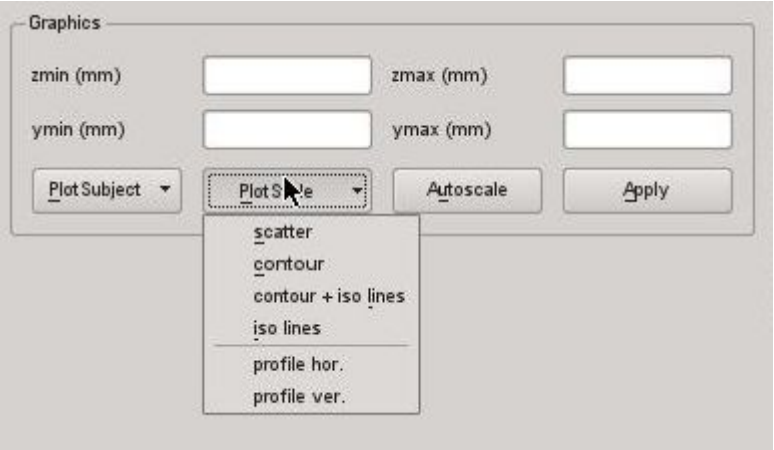


Figure 2: Footprint output of ray trace.

GRAPHIC OUTPUT





EXAMPLE - UNDULATOR PLUS KB MIRRORS.

Using an undulator source, the beamline has to be defined first including the wavelength before the undulator is defined. The wavelength is needed for the determination of the undulator size and divergence and an error occurs if the wavelength is equal to zero.

A vertically deflecting mirror 10m from the source, is used to focus the undulator radiation vertically 2 metres downstream of the mirror, and a horizontally deflecting mirror placed 0.5m beyond the first mirror, focuses horizontally in the same place. A slit is placed at the focal position. The continuation plane between the mirrors is set up half way between the mirrors. As an aperture is being included, a full ray trace has to be carried out (this includes apertures, slope errors, alignment errors) and the sizes of the optics have to be specified.

Select edit / beamline, and edit m1. In the optic definition form, set the preceding distance to 10000mm the succeeding distance to 250mm, the angle to 80, the source distance to 10000mm, and the image distance to 2000mm. Calculate the required radius and then set rho to the same value as r. To set the size of the optic in order not to clip any radiation set l1 and l2 (similarly w1 and w2) to -100 and +100 respectively in the lowest panel in the form. Make sure the up box is checked at the top of the form, hit apply and exit.

To add another optic after m1 in the beamline form, highlight the m1 file in the scroll box, then click on it so that the highlighting disappears, then press the <- key. Enter m2 in the file specification box. Pressing <- while the m1 file is highlighted, causes a new element to be inserted before m1.

For the second mirror, set the preceding distance to 250mm, the succeeding to 1500mm, the angle to 80 degrees, the source and image distances to 10500mm, 1500mm respectively. Again use the calculated value of r also for rho. Set the size as for the first mirror, and check the left button for a horizontally deflecting mirror. Apply and exit.

Add optical element m3 after m2, and set this to be an aperture. Set the distances to 0, The l and w settings are used to set the vertical and horizontal sizes of the aperture respectively. Set w1 and w2 to be -1.0 and + 1.0 respectively, and l1 and l2 to be -0.01 and +0.01.

In the beamline form, make sure the geometric optics box in checked and the wavelength is set. Hit apply and OK to generate the transform matrix for the whole beamline.

As the ray tracing is only at a single wavelength, only the source size and divergence of the radiation are needed, given by

$\sigma = \mathrm{sqrt}(\sigma_e^2 + \sigma_d^2)$ where σ_e is the contribution from the electron beam and σ_d is the diffraction limited source size or divergence given by

$\sigma_d = \mathrm{sqrt}(L\lambda)/4\pi$ for the source size, and

$\sigma_d = \mathrm{sqrt}(\lambda/L)$ for the divergence.

The electron beam parameters are hard wired into the code for BESSY and SLS undulators. In this example, select any of the predefined undulators (Figure 3) in the drop down list of the edit/source PSImage option, and specify the length and number of rays. Note the wavelength is greyed out and is taken from the input in the beamline box. There is also the option to select a generic undulator for which you need to enter the electron beam size and divergence and the undulator length.

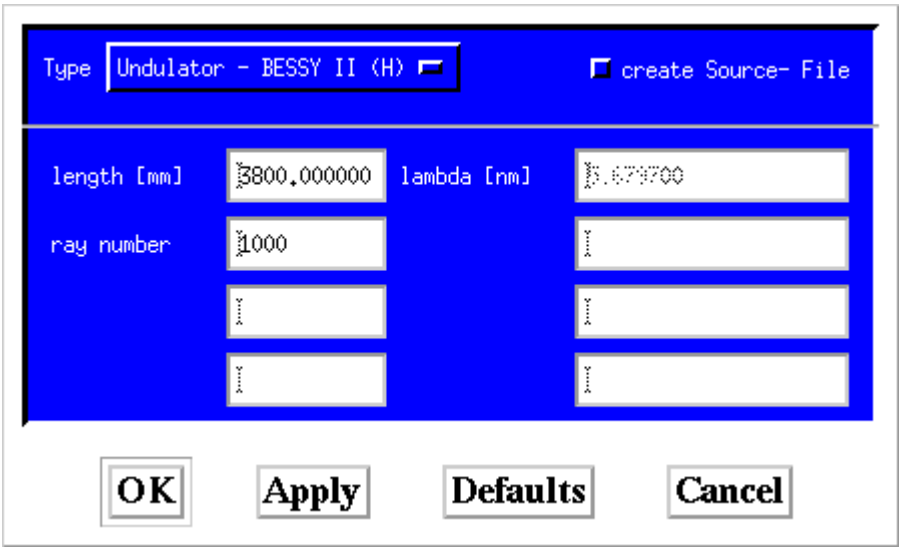


Figure 3: Undulator definition form.

Note, that if you go back to editing the beamline, you need to edit the source again because the wavelength might have been changed (otherwise an error is generated). In the future the code will be modified in a way that it will be necessary to re-edit the undulator source only if the wavelength has been actually modified.

Once the source is set up, select **full ray trace** in the calculation menu. If ray trace is selected, apertures, optics sizes and slope errors are not included. In the log of the code, it reports that 856 of the rays are lost, with a residual of 144 rays. Graphing the rays as before plots the distribution of the residual rays.

BEAMLINE OPTIMIZATION

OVERVIEW

Optimization of a beamline can be carried out using PHASE in conjunction with the code *phaseopti*, which uses the CERN library minimisation code MINUIT. The optimization part of the

PHASE package uses either *phaseqt* (PhaseQt gui) or phase (Motif gui) to set up a source and beamline. The optimization parameters have to be defined via the Motif gui. The information on the minimization is passed via 2 files, minuit.inp and test.pcko to *phaseopti* which then controls the minimization. The results are written to file, default name opti_out.dat.

In PHASE, the root of the name of *.pcko file is the same as the beamline file name. In order to change this to test.pcko required for *phaseopti*, change the name of the Opti Pickfile in the file/files option from the main menu.

Minuit requires a 'cost function' to minimise. The target to be minimized has to be defined in the Edit/Optimization window:

Target to be minimized	**Comments**	
Focus vertical	rms value	1
Focus, horizontal	rms value	2
Transmittance		3
Focus, horizontal and vertical	rms values	4
cost.F	see below	5
Resolving power, vertical		6
Resolving power, horizontal		7

The target cost.F has the meaning:

chi2= abs (ypc1(0,0,1,0) * dy + ypc1(0,0,2,0) *dy*dy +ypc1(0,0,3,0)*dy*dy*dy +

ypc1(0,0,0,2)*dz*dz + ypc1(0,0,1,2)*dz*dy*dy)

where ypc1(i,j,k,l) are the expansion parameters of the vertical displacement in the image plane with respect to the initial displacements and angles y, z, dy, dz. dy and dz are hardwired to 1.3 mrad and 0.32 mrad. The first three terms correspond to the well known terms defocussing, coma and third order aberration.

The cost function cost.F can be easily adapted to specific problems by editing and re-compiling the file cost.F. In the future the definition of the cost function can be done from the graphical interface.

Von der Bedienung hat sich folgendes geändert: in phase- edit- opti gibt es jetzt ein zusatzliches Menue "optimization target" bei dem man verschiedene Optimierungsmoeglichkeiten auswaehlen

kann- letztendlich wird eine zusätzliche Zahl in das file xyz.pcko aufgenommen (ev. alte files funktionieren weiter da das ueber den header erkannt wird). Das menue ist selbsterklaerend- die oberen 3 und die Eintraege mit res. power machen ein normales ray trace mit der aktuellen

Quelle und berechnen das entsprechende FWHM als chi. Transmittance macht das full-ray-trace und nimmt 1-transm als chi, focus special macht ein ray trace mit nur 4 strahlen - war einer meiner versuche- kann ev. auch wieder weg. cost.F ist die bisherige Variante mit neucompilieren- sollte nur noch selten gebraucht werden. Thomas sein Asphaerenfit konvergiert jetzt erstaunlich gut (Target FOCUS und Punktquelle) Startwerte und Schrittweite machen keinen grossen Unterschied. Den parameter "optimization target" kann man als 2. Parameter auf der kommandozeile uebrigens ueberschreiben (eine zahl 1...7).

In der .bashrc muss der pfad auf das Verzeichnis opti definiert sein. Dann kann man die Optimierung auch aus dem Menu heraus starten. Ansonsten kann man die Optimierung auch direkt aus dem Verzeichnis opti über die Kommandozeile aufrufen.

EXAMPLE - VARIABLE LINE SPACING GRATING

In PHASE, set up a hard edge source, using the default sizes for the distributions but including say 11 points in the size distributions and 17 in the angular. Set up a beamline consisting of a toroidal grating with 1200 lines/mm in 1st order, 20m from the source and focussing at 4m, as in Figure 4.

The line number of a VLS grating, counted from the center of the grating, is given by

$$N = N_0\, w + N_1\, w^2 + N_2\, w^3 + N_3\, w^4 + N_4\, w^5$$

The line density n follows from derivation with respect to w

$$n = N_0 + 2\, N_1\, w + 3\, N_2\, w^2 + 4\, N_3\, w^3 + 5\, N_4\, w^4$$

where w is the distance from the centre of the grating. The values of N_1 and N_2 will be found which minimise coma + defocussing.

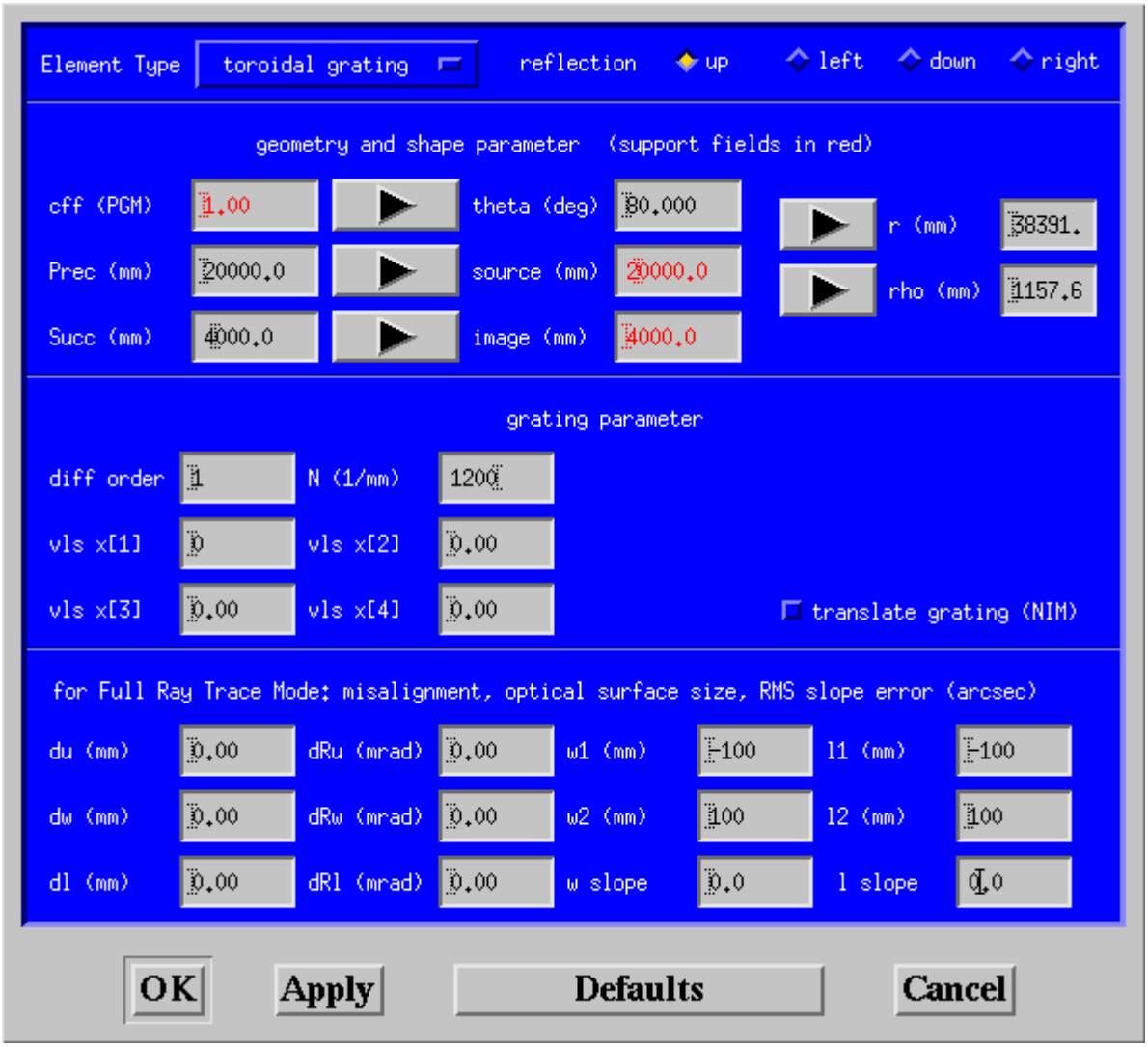


Figure 4: Parameters for a toroidal grating

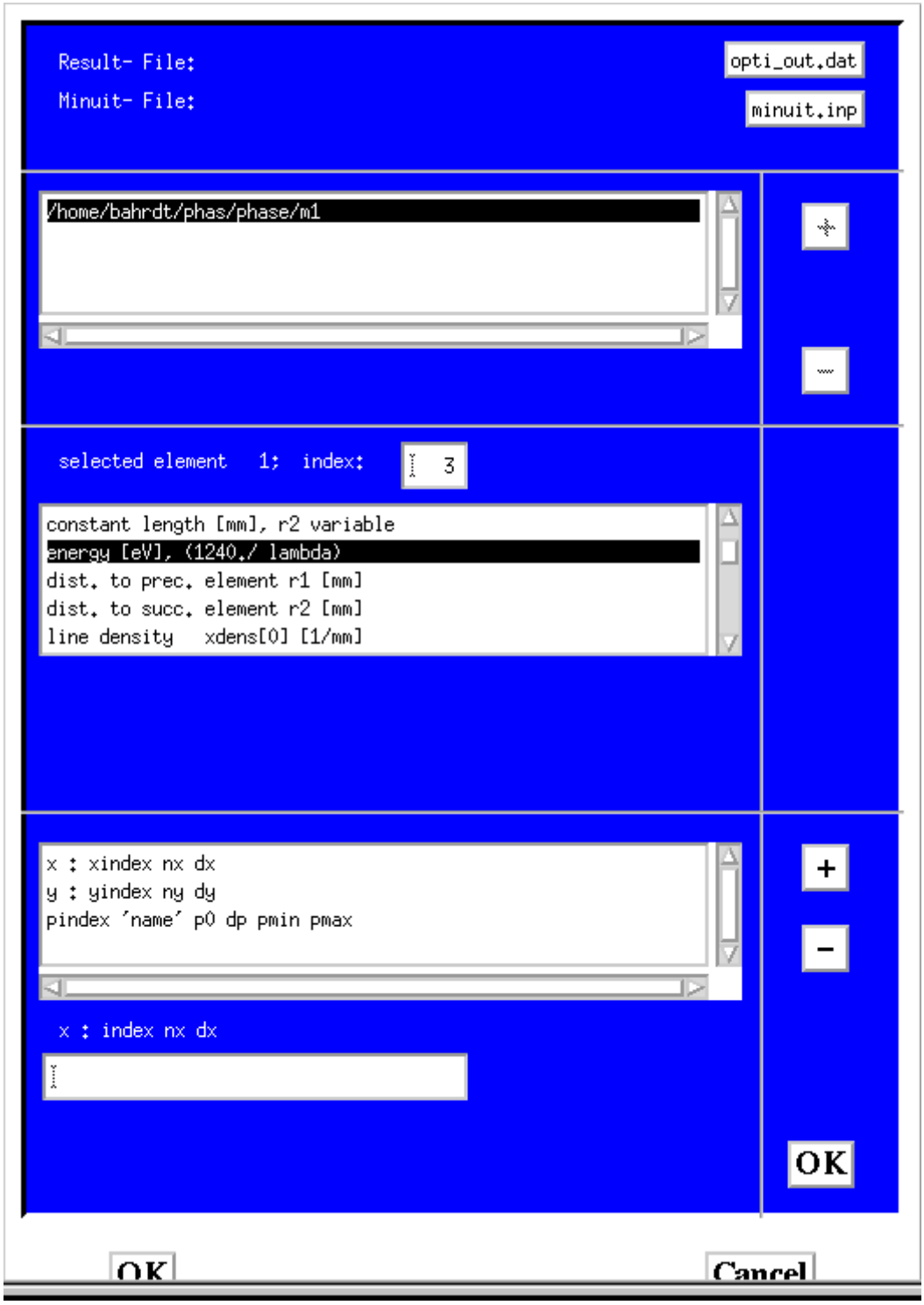


Figure 5: Form for specifying the minimisation.

Select edit/optimisation, to bring up the form Figure 5 and click on the optical element in the top box to highlight it.

Choose the target for optimization from the menu.

The minimisation can be carried out over a range of 2 different parameters, eg the coefficients for the equation governing the line density may be required over a range of energies and incident angles. The x and y parameters in the box in the lower section of the window describe these ranges. Note that the x and y ranges must be set even if you only want to carry out an optimisation for one set of parameters. The pindex line(s) describe the quantities to be varied in the minimisation.

The panel in the middle of the box gives the index for the parameters for the optical element highlighted in the top box. In this example, the linear and quadratic terms in the line density function will be found for 2 wavelengths, the value λ nm set in the beamline box and λ+2 nm and y will be the central line density.

The wavelength is index 11. Highlight the x: index nx dx line in the second bottom box, and then edit index nx dx in the bottom box so the line reads

x: 11 2 2

Click the upper OK box to accept the edit. Similarly, edit the y line to read

y: 6 0 1

The linear and quadratic terms for the line density are indices 7 and 8. Highlight the pindex line, and edit it in the bottom box to

7 'dens1' 0 1

(Note pmin and pmax do not need to be set unless the minimisation is going wild.) Highlight the pindex line in the second bottom and click + to get a new pindex line. Edit it to give

8 'dens2' 0 1.

Click the bottom OK button to create the required files for the optimisation.

There are two possibilities to start the optimisation:

1. phaseopti test.pcko

   the target for optimisation is defined in test.pcko

2. phaseopti test.pcko x

   x defines the target for optimisation and overwrites the value in test.pcko

Information on the progress of the optimisation is written to the screen and the log.

The file opti_out.dat contains the final values of x y chi2 optimised parameters. The results for the example are given below.

```
5.6797 1200 1.178035e-07 0.330478 -8.31414e-05
7.6797 1200 7.225151e-05 0.307068 5.15149e-05
```

Figure 6 and Figure 7 show the focus of the toroidal grating with constant line density and using the linear and quadratic terms in the line density equation, derived from the minimisation, for λ = 5.6797 nm.

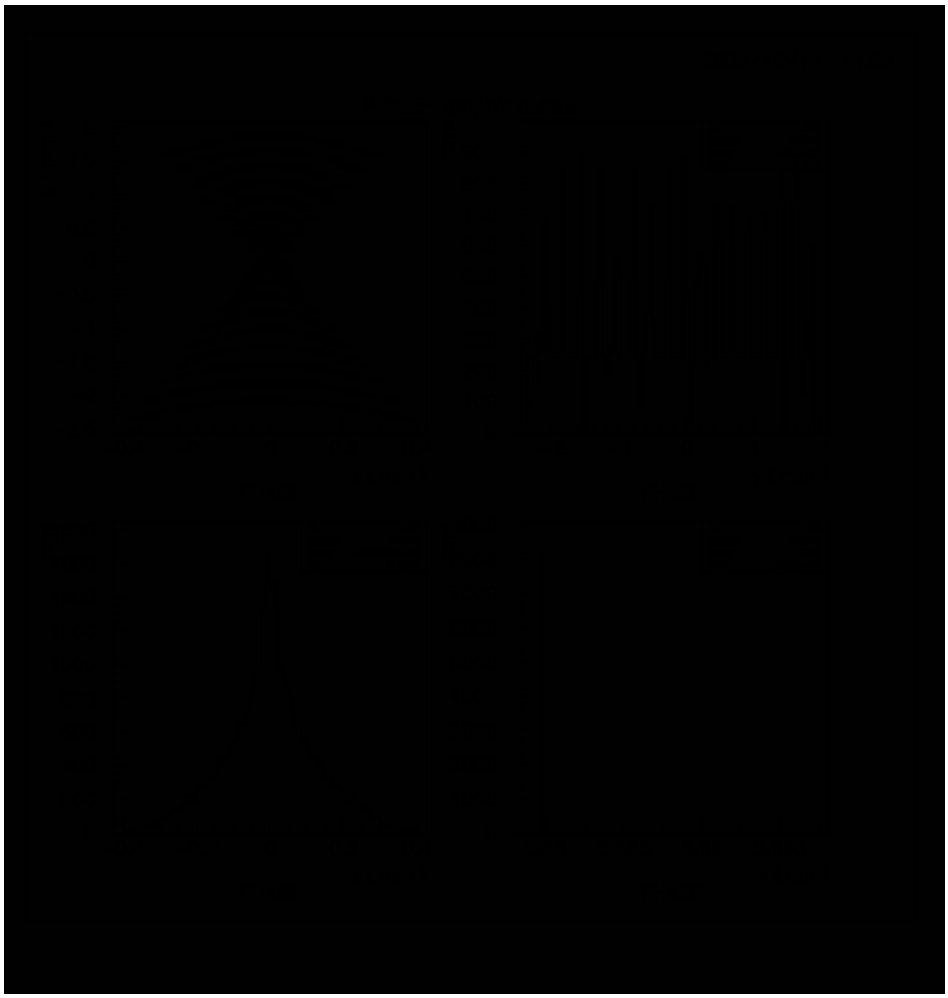


Figure 6: Focus of hard edged source using toroidal grating with constant line spacing of 1200 lines/mm

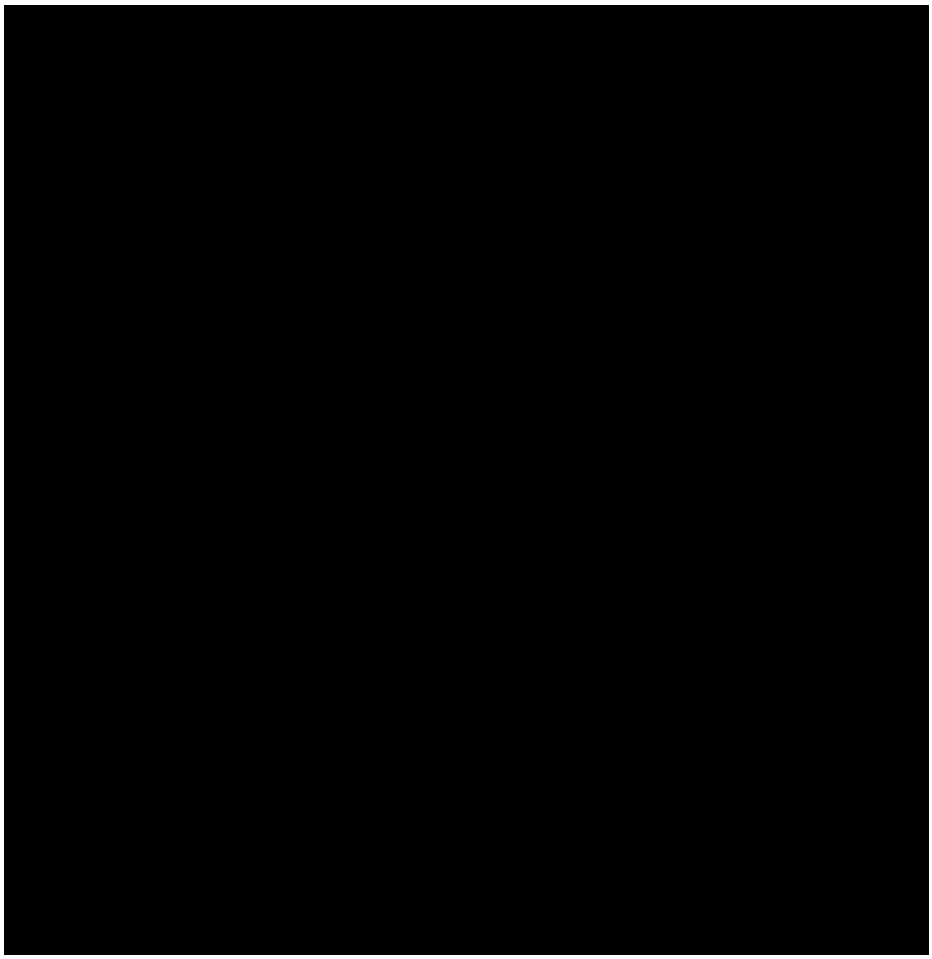


Figure 7: Focus of hard edged source using toroidal grating with variable line density grating.

PHYSICAL OPTICS PROPAGATION

OUTLINE OF METHOD

A very brief overview of the method is given to aid the understanding of how to set up these calculations. Please look into the listed references for the details.

The complex amplitude of the E field after propagation from an aperture a, via an optical element distance r to an image plane (a') at r' from the optical element is given by

$$E(a') = \int h(a,a') E(a)\, da$$

where

$$h(a,a') = 1/\lambda \int \exp(ik(r + r'))/(r\, r') \; (\cos\alpha + \cos\beta)/2 \; b(w,l)\, ds$$

b(w,l) is the reflectivity at the point w,l on the optic, α in the incident angle, β the reflected/diffracted angle, s is the surface of the optic.

I.e. the intensity for each point on the image plane is given by the double integral over all contributing positions ds on the optic and over all contributing position da from the source aperture.

The integral over s can be approximated to the sum of a set of amplitudes using the methods of stationary phase, and the expression for E(a') reduces to a single integral over da (= dxdy). The co-ordinates in the image plane are given by known 4th order expansions in the co-ordinates of the source plane. Hence the integration variables can be changed from the aperture plane positions to the slopes in the image plane, and the final expression can be written as

$$E(a') = 1/\lambda \int E(a) \exp(ik(r + r')) T\ b(w,l) \left| d(y,z)/d(dy', dz') \right| d(dy')d(dz')$$

where T is a set of scaling factors.

In PHASE, a grid of points is set up on the image plane. A set of angles is defined, and for each angle at each grid point, the contribution to the intensity from the source is found using the above expression. It is important that the mesh on the image plane, the range and number of angles is checked to be appropriate for the transformation, as will be shown in the example below.

GENERAL COMMENTS

Using the Physical Optics (PO) mode the following items are important:

- A complete asymptotic expansion up to 2nd order – equivalent to the stationary phase approximation - is implemented for one optical element (inorm2=3). Additionally, a 3rd order term in the OE coordinate w can be included which improves accuracy if critical points of the 1st kind are close and the contributions interfere (inorm2=4). The evaluation of several OEs can be done with a subsequent application of this algorithm (inorm2=3). A good approximation can be achieved with a different scaling factor (inorm2=0) which treats a whole set of OEs in a single step. The accuracy of this approximation should be checked once with a comparison vs. the method of subsequent propagation along the individual elements in the inorm2=3 mode. The propagation within a full 2nd order asymptotic expansion over a bunch of OEs in a single step is under development.

- PO propagation takes significantly longer than Ray Tracing (RT). PHASE permits an easy change between the two modes. It is recommended to set up the beamline including all optical elements in the RT mode and to switch to PO mode only if the beamline is set up correctly. Switching between modes requires only the adaption of the source parameters and the adaption of the distance to the 1st optical element.

- Due to the mathematics behind the code it must be avoided to have focal planes (or even one dimensional foci) simultaneously in the source and the image plane (this avoids division by zero). This is, however, not a severe constraint because the source field distribution can always be propagate in free space using FFT propagators of the PHASE4IDL package before starting the SPA propagation.

- The code propagates monochromatic light. Beams with finite bandwidths are described by frequency slices and each slice has to be propagated individually. There is a command option which permits multiple runs of phase with varying sources and wavelengths.
- The code propagates 100% coherent light. Incoherent light is easily handled with the RT option within Geometric Optics. The propagation of partially coherent light is not implemented. It can be emulated only by an incoherent sum of the results from a number of coherent light propagations. This has to be done manually.
- The source is defined under the menu point configure. The source is initialized automatically when starting the propagation.
- The propagation parameters are defined in the parameter box (bottom left).
- The source 'ps image plane' has to be used and the image size has to be defined.
- Misalignments of optical elements are included in physical optic mode.
- Slope errors in the mirror box are neglected in the physical optic mode. Only extended surface errors can be included via the 8^{th} order polynomial describing the optical element surface. Measured surface profiles can only be treated in an explicit (rather time consuming) integration of the Fresnel-Kirchhoff integral. This routine is included in the IDL part of the phase package. These simulations need a long CPU time.
- The optical element size as defined in the lower right box is not used in the physical optic mode. Apertures of optical elements have to be implemented via apertures as defined in the parameter box. If the aperture is located within the beamline the propagation has to be splitted and the 2^{nd} step has to be started at the aperture plane.

SOURCES

Currently, PHASE uses electric field distributions as source descriptions. This allows for simple and robust integration schemes. To speed up the simulations other (equivalent) representations may be more appropriate: An example is the amplitude and phase representation which usually needs much less data points. This option is under development.

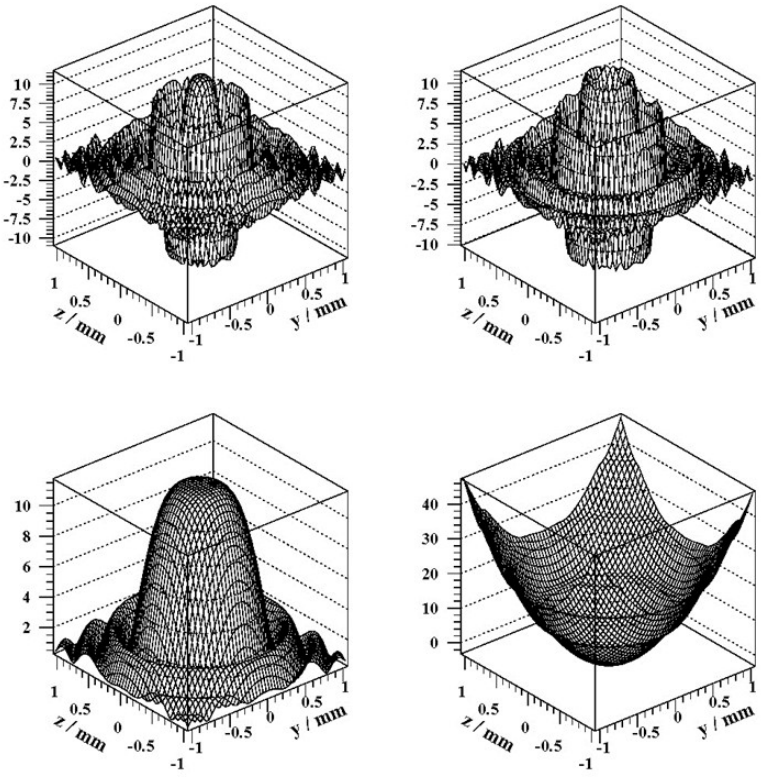


Figure 14 BESSY UE52 undulator at 105 eV described by real and imaginary part of the electric field (upper left and right) and described by ampplitude and phas (lower left and right) using the code PHACOR_SRC_NEW.

Electric field data from storage rings or FELs can be generated by various codes. We use WAVE for the simulation of undulators in 3rd generation storage rings and GENESIS for simulating FELs. There are several interface formats between these codes and PHASE:

WAVE FORMAT

This format can handle only one frequency slice which is represented by four individual files for eyre, eyim, ezre, ezim. Below are the first few lines of a typical file. The first line contains the number of z-points and the number of y-points. The following lines contain the z-position (first column), the y-position (second column) and 151 * 151 values of the real part of the y component of the electric field (last column). The units of the transverse co-ordinates (columns 1 and 2) are given in mm.

```
       151         151
-1.25690000000000       -1.25690000000000      -0.960800000000000
-1.24020000000000       -1.25690000000000       -4.09510000000000
-1.22340000000000       -1.25690000000000       -3.28110000000000
-1.20660000000000       -1.25690000000000       -1.41420000000000
-1.18990000000000       -1.25690000000000      -0.557920000000000
-1.17310000000000       -1.25690000000000        1.94460000000000
-1.15640000000000       -1.25690000000000       0.105910000000000
```

This format is used also in the PO multiple propagation mode, where the slices are treated one after the other. Here, PHASE reads in 4 files for each new frequency slice.

HDF5 Format

This format can easily handle several frequency slices. Currently, PHASE propagates only one slice, but is intended to extend the code for the propagation of several frequency slices in parallel. A number of slices will be included in one single file. The following routines for data conversion are available:

- **Conversion of genesis file to phase hdf5 file:**
  genesis2phase_hdf5 **?????**

- **Conversion of wave files to phase hdf5 file:**
  wave2phase_hdf5 list_of_slices outputfile genesis2phase_hdf5
  one slice consists of time value and 4 files representing eyre, eyim, ezre,ezim
  example for 2 slices:
  wave2phase_hdf5 0.5 eyreal1 eyimag1 ezreal1 ezimag1 1.0 eyreal2 eyimag2 ezreal2 ezimag2 output.hdf5

- **Conversion of wave files to genesis hdf5 file:**
  wave2genesis_hdf5 list_of_slices outputfile
  one slice consists of time value and 4 files representing eyre, eyim, ezre,ezim
  example for 2 slices:
  wave2genesis_hdf5 0.5 eyreal1 eyimag1 ezreal1 ezimag1 1.0 eyreal2 eyimag2 ezreal2 ezimag2 output.hdf5

Starting these routines without parameters they print a usage message on the screen.

Apertures

Optical Elements

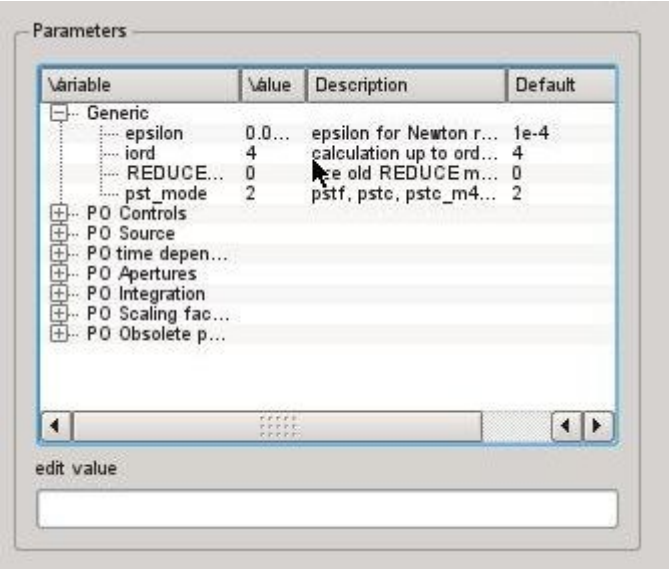
The optical elements are defined in the same way as for RT Optical Elements with the exceptions:

- The slope error definition is not applicable
- The optical element size definition is not applicable

Integration Parameters

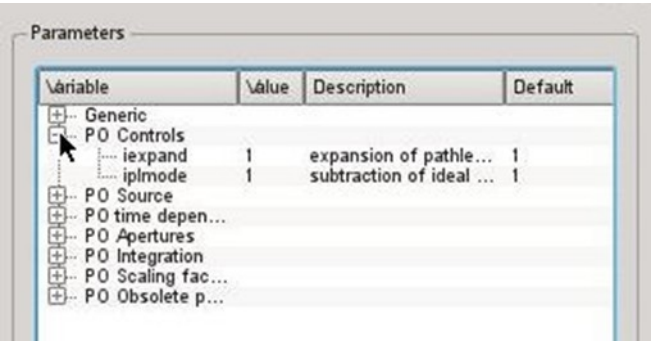
The parameters are defined in the parameter field at the lower left on the main window.

Generic Parameters



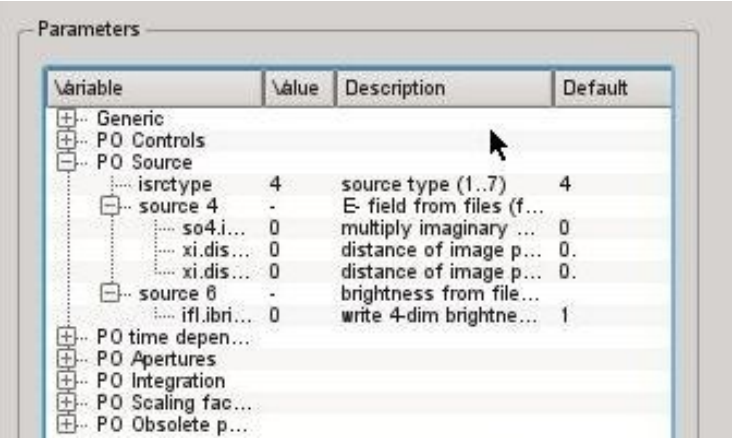
- *epsilon:* The code is based on lots of power series expansions. The algebra with power series repeatedly uses the Newton Raphson method. *-epsilon* defines the accuracy of these evaluations. 1e-4 is usually sufficient.
- *iord:* order of the power series expansion. The old REDUCE code is limited to *iord=4*, the new routines can be used up to *iord=7*. The maximum value is equal to the surface shape expansion order minus one.
- *REDUCE:* The parameter switches between the old REDUCE code (1) and the new code (0) as described in the SPIE paper 2011.
- *pst_mode:* The parameter switches between the old fortran integration routine pstf.F (0) and the new routine pst.c (2). (1) is used for debugging purposes only. *pst_mode=0* should be used for automated multiple frequency simulations within one *phaseqt* session. Otherwise, *pst_mode=2* should be used.

PO CONTROLS



- *iexpand:* (1) the path length is expanded up to order iord. This is the usual setting. (0) The exact path length is evaluated (for debugging and for one optical element only).
- *Iplmode:* (1) the averaged path length is subtracted analytically to avoid simulations with large numbers with many significant digits. This is the usual setting. (0) for debugging only.

PO SOURCE



Two PO source types are currently implemented (4 and 7) and a brightness source is under development (6).

Source 4: The source is defined in the WAVE Format. The real and imaginary part of the horizontal and the vertical electric field amplitude is defined in four separate files, e.g. eyreal.dat eyimag.dat ezreal.dat ezimag.dat.

Source 7: The source is defined in hdf5 format HDF5 Format.

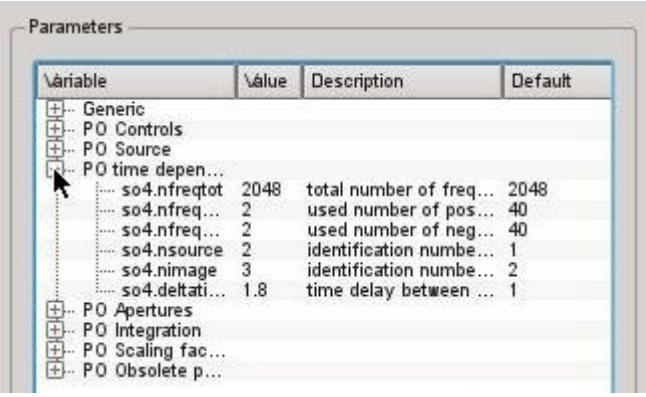
Source 6: Another approach for a source description is the brightness (K.-J. Kim) or Wigner distribution. This 4-dimensional quantity (depending upon two coordinates (y,z) and two angles (y',z') does not correspond to a physical quantity. It is not positive definite. Integration over two variables gives a measurable quantity, e.g. the transverse distribution vs. (y,z) or the divergences (y',z'). This source type is under development.

In case the image plane is out of focus the transverse co-ordinates in the image plane and the angles for which the major contributions to the integral appear are strongly correlated. To avoid adding lots of zeros one should set the parameters distfocy and distfocz (distances from the image plane to the vertical and the horizontal focus) appropriately. distfoc is positive if the image plane is located downstream to the focus. Then, at the transverse positions y' and z' in the image plane the angles of integration are centred around the angles

dy'0 = y' / distfocy and dz'0 = z' / distfocz

Be aware that the divergencies dy' and dz' which have to be integrated are much smaller out of focus than in the focus. Out of focus the required divergencies are roughly dy'=vertical focus size/distance from focus and similar for dz'.

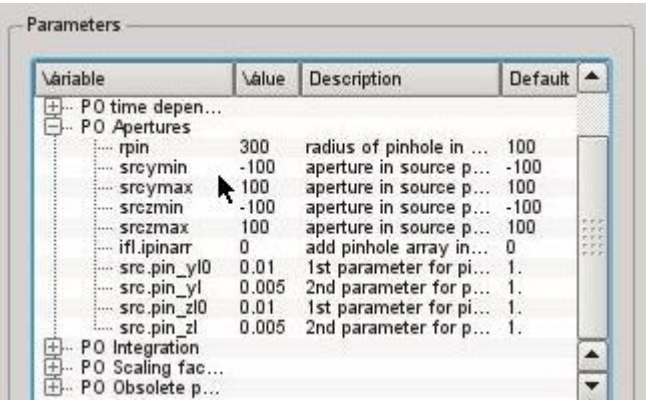
PO Multiple Propagation



FEL codes such as GENESIS are used to generate time dependent light pulses. With FFT operations the time slices can be transformed into frequency slices. This step has to be performed outside from the monolithic SPA phase. In the time dependent mode phase reads one slice after the other, propagates each slice and write output files. Input files and output files are organized in WAVE format. The following name convention is expected: EYRExxxx.day EYIMxxxx.day etc., where xxxx is the number of the frequency slice and y is the ID indicating source and image. Positive frequencies are running from *0 - nfreqtot/2* and negative frequencies from *nfreqtot/2+1 – nfreqtot.*

- *nfreqtot:* total number of frequency slices as generated from GENESIS time dependent file.
- *nfreqpos:* number of used positive frequencies (slice numbers: *0 to nfreqpos-1*)
- *nfreqneg:* number of used negative frequencies (slice numbers: *nfreqtot to nfreqtot-nfreqneg+1*)
- *nsource:* ID of source slice (one digit number)
- *nimage:* ID of image slice (one digit number)
- *deltatime:* time delay between two time slices

PO APERTURES

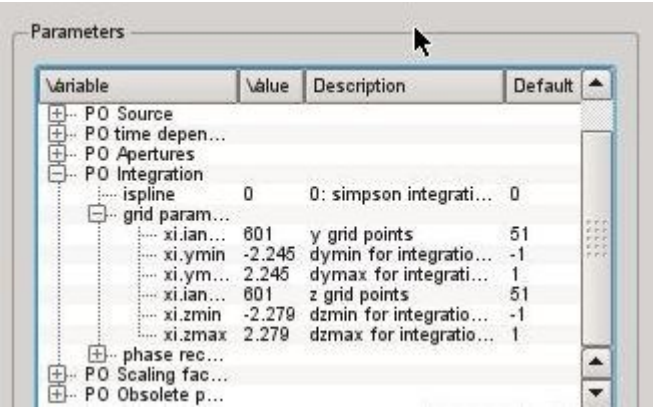


Apertures can be applied to the source field distribution. To propagate through an aperture, which is not located in the source plane, it is necessary to proceed in 2 steps i) propagate the field distribution to the aperture plane and save the results to files; ii) These files then serve as the input to a further propagation, where the desired apertures are applied prior to this step.

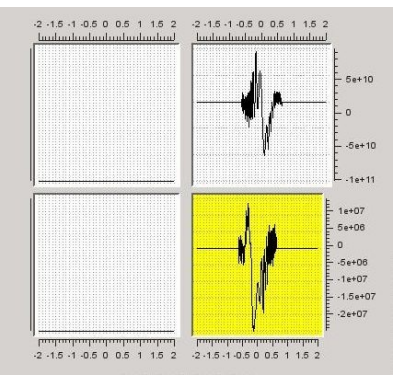
- *rpin:* radius of circular aperture
- *srcymin, srcymax:* vertical aperture limits
- *srczmin, srczmax:* horizontal aperture limits

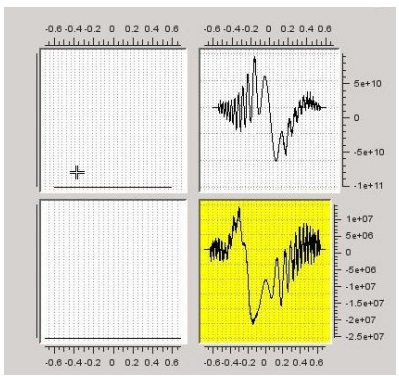
- *ipinarr:* (1) a pinhole array aperture is enabled; (0) no pinhole array aperture
- *the following 4 parameters describe the micro apertures (size and distances)*

PO INTEGRATION



- *ispline = 0:* Simpson integration; this is a robust method which does the work for a sufficiently high density of angles. The four electric field files eyre, eyim, ezre, ezim describe the real and imaginary part of the vertical and horizontal electric field vector in the source plane.
- ispline = 1: Cubic spline interpolation; currently only equidistant grid implemented
- ispline = 2: Cubic spline interpolation; the data are sorted for increasing x-values
- *ispline < 0 (under construction, not available, yet):* For rapidly oscillating integrands, the Simpson integration may require a large number of integration steps. A more sophisticated integration scheme is based on an amplitude and phase representation of the source. Since the fluctuation of the integrand may be much smaller an analytic integration can be performed which reduces the number of required integration steps significantly. This method is not suited for noisy data where an analytic description of parts of the fields is inappropriate. Furthermore, the two dimensional transformation from fields to amplitudes and phases is difficult and there is no robust algorithm available, currently.
- *grid parameters, ranges:* The range of integration, i.e. ymin / ymax and zmin / zmax for the vertical and horizontal plane, have to be minimized in order to avoid useless summation over zero intensity. The integration angles are optimized by pushing the button autorange (Figure xx). This button gives useful results only if the numbers distfocy / distfocz are set correctly.





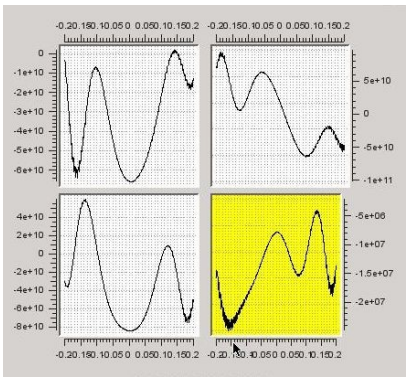


Figure xx: Plots of simpre, which show various contributions to the field at the point selected in the image plane. The top left plot shows the contribution from each dy'dz' element as a function of y' for z'(min), the lower left plot for z'(max) and the top right plot for the mid value of z'. The lower right hand plot shows the integrated contributions over y' for each z'. Left: Integration ranges ymin / ymax and zmin / zmax are unnecessarily large. Center: Ranges have been determined with autorange button. They are well adapted. Right: Ranges are too small and information from the outer part of the source is lost.

- *grid parameters, number of steps:* The computing time is directly proportional to the number of angles ianzy (vertical) and ianzz (horizontal) and, therefore, these numbers should be kept as low as possible without sacrificing accuracy. If the numbers are too small aliasing effects may occur (Figure xx). The numbers are appropriately chosen if they are set to the minimum value where the result (2-dimensional distribution in the image plane) does not change when doubling the step numbers.

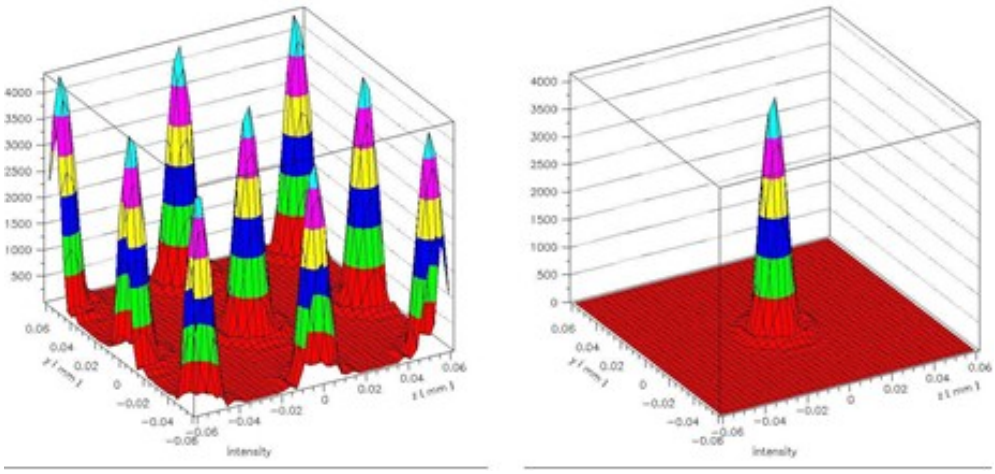
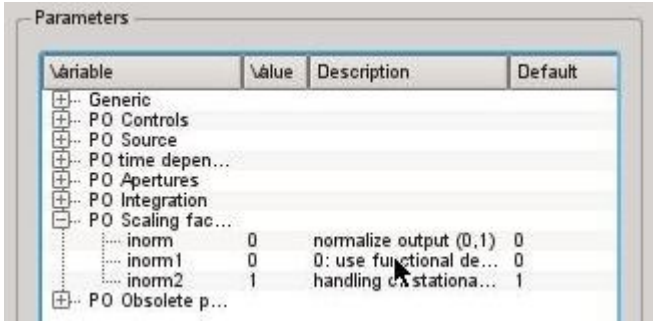


Figure xx: Left: The numbers of integration steps are too small. Right the numbers are sufficient, though, they may be too large.

PO Scaling Factors



- *inorm:* (1) the output intensity distribution is normalized to one; (0) the output is not normalized
- *inorm1:* (0) the determinant is evaluated accuartly; (1) the determinant is set to one. This mode is used for debugging, only.

- *inorm2:* This is the key parameter for the SPA algorithm. It defines the way how the analytic integration over the optical element surface is performed (see Appendix A).

OUTPUT

GRAPHIC OUTPUT

FILES

EXAMPLE - UNDULATOR SOURCE IMAGED BY A TOROIDAL MIRROR.

The input files used (eyre.dat etc) were generated by the program WAVE of M. Scheer, and give the complex electric field at 10m downstream from the undulator. A toroidal mirror 20m from the undulator focuses the light at 24m from the undulator.

Note that it is recommended to ray trace the system first to check it is set up correctly, before carrying out the propagation.

The steps required for the propagation are

- Set up (or amend) the beamline description
- Set up a small grid of points on the image plane
- Set up input field files, angles ranges etc, defined in fg34.par
- Carry out the propagation
- Check the angle range is OK
- Carry out propagation for full set of image grid points when the set-up is OK.

In Edit beamline, define a single optical element. In the optical element definition form, select and upward facing toroidal mirror, with the preceding plane at 10 m (the distance from the plane at which the electric field has been calculated), the distance to the next plane 4m, the angle 80 degrees, source and image distances of 20m and 4m respectively, and use the r and rho arrow boxes to calculate the mirror radii, and apply. In the beamline box, enter the wavelength at which the field has been calculated, check the **physical optic box** (for Phase Space Imaging) option, then apply and OK.

Edit fg34.par to set the file names for the input fields, eg eyre.dat, eyim.dat, eyze.dat, eyim.dat for the real and imaginary parts of the y and z fields. The other parameters can either be set editing the file, or else using the read fg34.par followed by edit parameters options in the edit menu of the code. In the latter method, the values of the parameters are changed in the code, but the values in the file are not altered.

Check and set if necessary the following parameters from fg34.par

- epsilon for Newton routine: 1.e-4

- source type: 4
- Subtraction of ideal pathlength : 1
- Order of calculation : 4
- Simpson integration or spline : 0 (for Simpson) (but see section 4.2.1)

Set the range and grid of angles for propagation back from the image points using

- Ymin, ymax, ianzy0,
- Zmin zmax, ianzz0,

for the range and number of angles in the y and z directions. The suggested starting values are -1.5, 1.5, 201.

After reading in fg34.par, select init source from the edit menu to read in the electric field values.

From the edit source/ps Image option select PS Image plane. Set up the grid of points for the image plane (Figure 8). As the calculation can be quite lengthy, initially set up only a single row of points. Choose ymin and ymax and number of y values appropriately. In the z-plane use zmin=zmax=0.0001 and number of a points=2 (number of points = 1 will cause problems during plotting). Hit apply and OK. Then under command/calculation select phase space imaging.

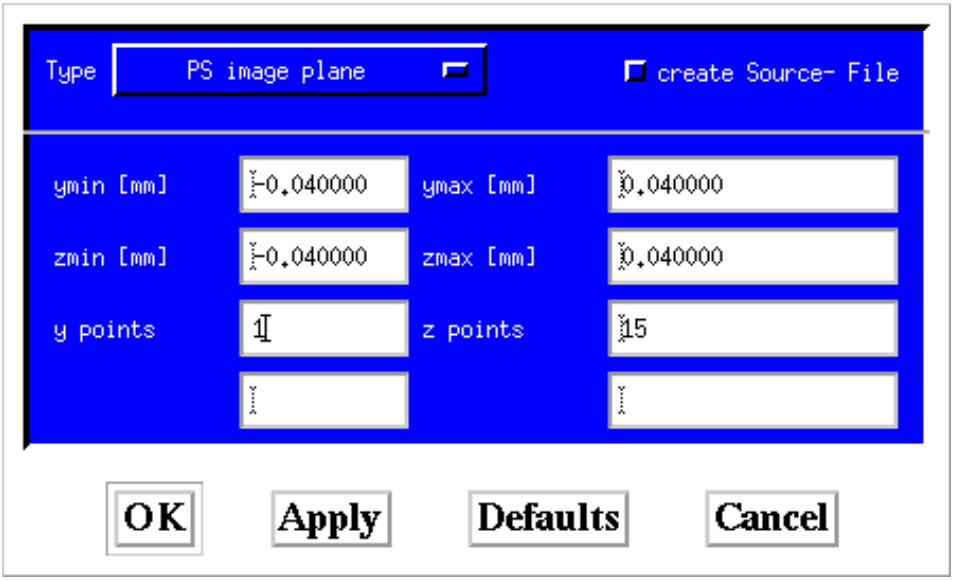


Figure 8: Form to define a grid in the image plane for wavefront propagation.

TIME DEPENDENT CALCULATIONS

PHASE provides the option to do PO multiple propagation in a loop. In this mode the information of a complete light pulse including different frequencies can be simulated. Time dependent FEL simulations with e.g. GENESIS produce electric field distributions (transversally distributed) for many time slices. To derive a frequency representation from the GENESIS output FFTs are applied to each grid point of the transverse distribution. Each frequency slice can now be

propagated with PHASE. Not all frequencies have to be used but only those which have a significant intensity. The number of frequencies to be used has to be larger in the SASE case as compared to a HGHG case. The total number of frequencies (NFREQTOT) produced in the FFT and the number of frequencies to be propagated in PHASE (NFREQPOS and NFREQNEG) has to be defined in FG34.PAR. Also the time difference between two subsequent slices in the time domain has to be defined in FG34.PAR (units are fsec).

The File name convention for the frequency slices is as follows:

Transverse distributions of the electric field components:

EYRESxxxxx.DAy, EYIMSxxxxx.DAy, EZRESxxxxx.DAy, EZIMSxxxxx.DAy

xxxxx is a five digit number representing the individual frequency. xxxxx=1 is the nominal frequency defined in the beamline window. xxxxx=2 is the next higher frequency and NFREQTOT the next lower frequency with respect to the nominal frequency.

y is a one digit number identifying the source and image file, respectively. The number y for the source and the image file is given in FG34.PAR.

The command “multiple phase space imaging” starts the loop over the PHASE calculations. For each frequency one run is performed. PHASE propagates the frequencies 1…NFREQPOS and NFREQTOT-NFREQNEG+1…NFREQTOT. The output data have the same structure as the input data. The frequency slices of the output data can be converted via FFT-1 back to a time dependent representation. These files can then be used in subsequent GENESIS calculations.

Examples for this method are given in the FEL2005 proceedings.

BATCH MODE & PARALLEL PROCESSING

PO-propagation, and particularly time dependent simulations, may require lots of computation resources and parallel processing is strongly recommended. Various methods for paralle computing are implemented.

The code PHASE is capable to use many cores of a single computer. The code has been tested on a machine with 64 cores. Within the interactive *phaseqt* multi-threading is involved by choosing the “sun” button. Single CPU simulations are started with the “PO” button. A thread pool is provided automatically by qt where the number of threads depends upon the hardware architecture (number of CPUs) and the system workload. The individual points $(i, j) \forall i, j : 1 \le i \le n, 1 \le j \le m$ in the image are evaluated in parallel. A task pool is defined with one task per point. qt organizes the distribution of the tasks among the available threads.

Alternatively to the interactive mode, PHASE can be started in batchmode. Once, the beamline file xyz.phase has been setup via the phaseqt interface or by editing the beamline file the propagation can be started in batch mode with following command:

phaseqt –b -m6 –O2 –t16 xyz.phase

The server version is started with

phasesrv -m6 –O2 –t16 xyz.phase

The parallelization in the batchmode and the server mode is based on the posix thread library. The number of threads has to be specified manually.

-mx: -m6 indicates parallel processing, -m3 is used for single task (single cpu) simulations. *phasesrv* does not utilize all available threads automatically, and the number of threads has to be specified, instead. The number of available threads depends upon the hardware and the workload on the system. It is limited to the number of CPUs of the system.

-tx: The number following "t" is the number of threads to be used. The default value is 4. With the option –tx the evaluation of the points $(i, j) \forall i, j : 1 \le i \le n, 1 \le j \le m$ in the image plane is distributed among x threads where each threads handles $m \cdot n / x$ data points.

- Ox: The output format is defined with –Ox :

- O1: WAVE compatible output (5 files)
- O2: PHASE hdf5 file
- O3: genesis hdf5 file

Both hdf5 formats include the fields and the intensities in one single file.

unter src/phaseidl/ habe ich paar idl scripte plothdf5* mit denen man die hdf5 files plotten kann - am einfachsten mit dem wrapper script plothdf5 das man mit filenamen und einem switch aufruft, Bsp.:
plothdf5, 'filename.h5', /psd, /png

wuerde die Intensitaet plotten und zusaetzlich als png file speichern, alternative Optionen /genesis und /phase plotten
die Felder je nach filetyp - dieser wird in den scripten noch nicht automatisch erkannt

Multiprocessing
real multiprocessing with OpenMPI:
parallelization within c-routine: fkoe.c auf: ~/phase_0/philipp/net/home/hid/dirac/fkoe/src
pseudo multiprocessing in a Pearl script
parallelization within script, no synchronization: Parallel_Phase_Pearl

Multithreading
Low-level parallelizing (do-loops): fkoe in Parallel_IDL_OpenMP
High-level parallelizing (function pst_i): phaseqt von Uwe
Very basic parallelizing without specifc libraries (server version): parallel_PThreads

APPENDIX A: SPA-PARAMETERS

inorm1 (0) factor1 = $\left|\frac{\partial(y,z)}{\partial(dy',dz')}\right|$ (normal case)

(1) factor1 = 1.0 (for debugging, only)

inorm2 the preferred selections are marked in red

(0) factor2 = $\left|\frac{\partial(y,z)}{\partial(dy',dz')}\right|^{-0.5}$ (for debugging, only)

(1) factor2 = 1.0 (for debugging, only)

(21) $$factor2 = 2\pi \frac{\sqrt{\cos(\alpha)\cdot\cos(\beta)}}{rr'}\left|\frac{\partial^2 PL}{\partial \Delta w^2}\cdot\frac{\partial^2 PL}{\partial \Delta l^2}\right|^{-0.5}$$

(2) Taylor Series expansion of the expression (21)

(31) $$factor2 = 2\pi \frac{\sqrt{\cos(\alpha)\cdot\cos(\beta)}}{rr'}\left|\frac{\partial^2 PL}{\partial \Delta w^2}\cdot\frac{\partial^2 PL}{\partial \Delta l^2}-\left(\frac{\partial^2 PL}{\Delta w\cdot\Delta l}\right)^2\right|^{-0.5}$$

No Taylor series expansions of products, sqrt, inverse

(32) coordinate transformation (version 1) of values (not power series) for cross term removal (for debugging, only); same result as (31)

(33) coordinate transformation (version 2) of values (not power series) for cross term removal (for debugging, only); same result as (31)

(3) Taylor series expansion of denominator of the expression (31); less accurate results

(4) factor2 = lookup table as generated with Mathematica; incl. 3rd order term in w

Ausdruck von Isabelle für 3. Ordnungsterm according to convlim

(40) $$factor2 = 2\pi \frac{\sqrt{\cos(\alpha)\cdot\cos(\beta)}}{rr'}\left|\frac{\partial^2 PL}{\partial \Delta \tilde{w}^2}\cdot\frac{\partial^2 PL}{\partial \Delta \check{l}^2}\right|^{-0.5}$$

$\tilde{w}, \check{l}$ are transformed coordinates for removal of cross term (version 1)

(41) $$factor2 = 2\pi \frac{\sqrt{\cos(\alpha)\cdot\cos(\beta)}}{rr'}\left|\frac{\partial^2 PL}{\partial \Delta \tilde{w}^2}\cdot\frac{\partial^2 PL}{\partial \Delta \check{l}^2}\right|^{-0.5}$$

$\tilde{w}, \check{l}$ are transformed coordinates for removal of cross terms (version 2)

(43) factor 2 = equation of Isabelle; inlc. 3rd order term in w

(44)

(45)

(5) factor2 = includes complete asymptotic expansion (not implemented, yet)

REFERENCES

- Fourth order optical aberration and phase-space transformation for reflection and diffraction optics - J. Bahrdt, Appl Opt. 34, 114 (1995)
- Beamline optimization and phase space transformation - J. Bahrdt, U. Flechsig and F. Senf, Rev. Sci. Instrum. 66, 2719 (1995)
- Wave-front propagation: design code for synchrotron radiation beam lines - J. Bahrdt, Appl Opt 36, 4367 (1997)
- Wave front propagation in synchrotron radiation beamlines - J. Bahrdt, U. Flechsig, published in "Gratings and grating monochromators for synchrotron radiation", Proc. of SPIE, Vol. 3150 pp158-170, San Diego, 1997
- Wavefron Tracking – J. Bahrdt, Proceedings of the FEL 2005 conference, Stanford, Ca, USA
- J. Bahrdt, “Wavefront tracking within the stationary phase approximation”, PRST AB, 10, 060701 (2207).

  http://prst-ab.aps.org/abstract/PRSTAB/v10/i6/e060701
- PHASE, a Universal Software Package for the Propagation of Time-Dependent Coherent Light Pulses along Grazing Incidence Optics - J. Bahrdt, U. Flechsig, S. Gerhardt, I. Schneider, Proc. of SPIE Vol. 81410E pp1-10, San Diego, Ca, USA, 2011.
- Propagation of Coherent Light Pulses with PHASE - J. Bahrdt, U. Flechsig, W. Grizolli, F. Siewert, Proc. of SPIE Vol. 9209 pp920908-1-18, San Diego Ca, USA, 2014.