McStas Simulation for PANDA
Part Ia: Primary Spectrometer -

M. Rotter & R.Schedler
Institut für Angewandte Physik, Technische Universität Dresden
D-01062 Dresden, Germany
E-mail: rotter@physik.tu-dresden.de

April 23, 2001

Contents

1 Introduction

This report describes the Monte Carlo Simulation which was carried out for the triple axis spectrometer PANDA using the `McStas` Software (see `http://elu-alf-2.risoe.dk/~elu-krni/mcstas` for details).

The guide was taken as designed at 6.okt 99. Changes made hereafter have not been included in the calculation (for details please refer to the geometry given in the program listing in the appendix). For the supermirrors a reflectivity of 75% was used. The absorption for the primary collimator (C1) was taken to be 5% for 60min, 9% for 40 min and 15% for 20min.For the other collimators (C2,C3,c4) the absorption values of 2% for 60min, 3% for 40min and 5.5% for 20min were assumed.

The total flux at the sample was calculated using the formula:

$$tot1x1spl \left[\frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2}}\right] = flu1x1spl \left[\frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}\right] \cdot \frac{dE\,[\mathrm{meV}\,]}{2 \cdot E\,[\mathrm{meV}\,]} \cdot wavelength\,[\mathrm{\AA}] \tag{1}$$

Energy Monitors: the plot show the neutrons per second for each energy channel. In order to get the neutron flux per seconds, area and energy it is necessary to divide these values by the actual energy channel width and detector area:

$$spectral\ neutron\ flux\ density \left[\frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}\right] = \frac{intensity\ on\ energy\ monitor\ \left[\frac{\mathrm{n}}{\mathrm{s{\cdot}(energy\ channel)}}\right]}{detector\ area\,[\mathrm{cm^2}] \cdot channel\ width} \tag{2}$$

Note: no filters have been taken account of ! The sensitivity of the detectors has not been taken into account !

2 Simulation of the neutron source

In order to dimension the neutron shield surrounding the beamlines some calculations about the maximal neutron flux were done by Dr. Gaubatz (FRM II). From these simulations we using the informations about the flux of cold neutrons at the beamline SR2. The original calculations describing the flux in a detector at a distance of 400 cm from the reactor core axis. For our calculations we need the neutron flux at the starting point of our beam tube.

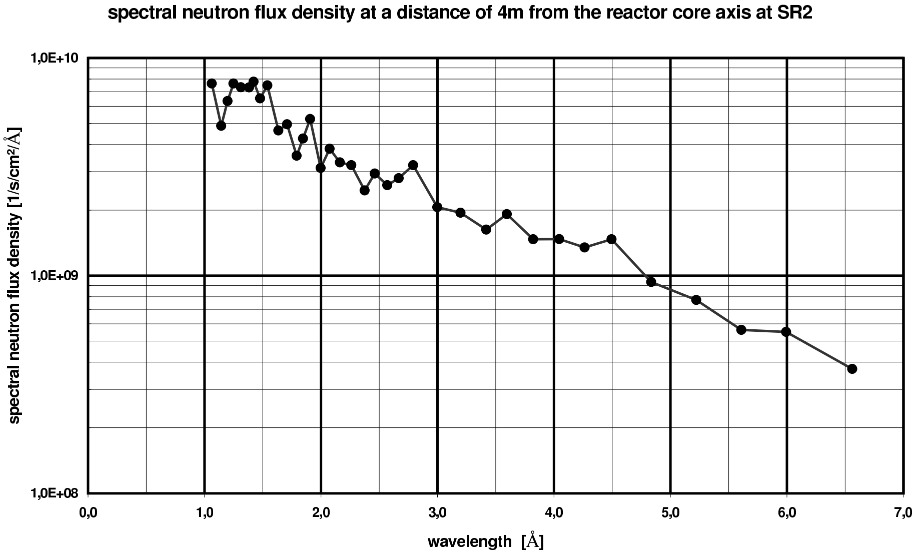

Figure 1: cold neutron flux given by Dr. Glaubatz (FRM II)

In order to get this flux we simulated a neutron source [1] at the position of the cold neutron source (distance reactor core axis to cold neutron source is 0.4 m). Then we switched the source on and de-

[1] For the simulation of the neutron source the "unofficial" `McStas` component Source_flux.comp was used. This component is a variant of the official Source_flat component and models a reactor source with a flat energy distribution and a given

termined the flux on a virtual Detector at distance of 3.6 m from this source. At next we tuned the source flux to a specific value, so that the "measured" flux was the same as in the given calculations (Dr. Glaubatz). In this manner we achived a neutron flux of $3.78{\cdot}10^9\ \frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}$ at the virtual Detector position provoked by a source flux of $4.5{\cdot}10^{12}\ \frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}$.
Some futher corrections are necessary because of the wavelength dependence of the Neutron flux. The wavelength dependence of the cold neutron source can be described by the following approximation:

$$\textit{spectral neutron flux density}\left[\frac{n}{s{\cdot}cm^2{\cdot}\AA}\right] = e^{\,(30.3-0.53\cdot\lambda[\AA])} \tag{3}$$

All McStas simulations were done with the same initial neutron flux of $4.5{\cdot}10^{12}\ \frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}$. Every resulting flux in our calculations must be multiplied by a given wavelength dependent factor. The following table was created from the previous equation and present the most common correction factors.

wavelength [Å]	6.2	5.0	4.0	2.4	2.2	1.5
flux $\left[1\cdot10^{12}\cdot\frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}\right]$	0.54	1.02	1.73	4.04	4.50	6.51
correction factor	0.12	0.27	0.38	0.90	1.00	1.45

3 Program Operation

3.1 Basic Simulation

To perform the calculations the spectrometer was mapped to a McStas program. The program consists of a main module panda.ins and several sub modules which are called by the main module (for a complete listing see appendix B).

The program can be compiled using the McStas compiler and accepts several variable parameters for the simulation, like initial and final energy of the neutron, diaphragm width, scattering vector, etc. In the real spectrometer these paramters will be controlled by the spectrometer software. The output of the simulation can be viewed via the mcplot and mcdisplay commands.

3.2 Changing the Spectrometer Modes

The Spectrometer will be designed to operate in different configurations using different equipment, like several monochromator crystals, diaphragms, collimators, etc. The task was to change the simulation program according to the different configurations which are possible during the normal operation.

Therefore the program was written to operate in the following 5 principle modes: [2]

MODE 1: shutter-revolver No. 1, horizontal and vertical focussing mode,no collimators

MODE 2: shutter-revolver No. 2, focussing Heusler mode (not fully implemented yet (no polarized neutron simulation is possible in McStas up to now) C1=20,40,60min, between Heussler monochromator and sample there is inserted a convergent neutron guide

MODE 3: shutter-revolver No.3, horizontal collimated mode, Collimators C1,C2,C3,C4=20,40,60 inserted, vertical but no horizontal focussing at both Monochromator and Analyzer,

neutron flux. This is useful for simulations where the absolute value of neutron flux, detector counts, etc. is needed for comparison with real instruments and experiments.

[2] modes can be easily switched in the program by changing in all *.ins files the comments of the type /*m1*/ ... /*1m*/ into /*m1 ... 1m*/ to switch off mode 1, and changing the comments of (for example) /*m2 ... 2m*/ into /*m2*/ ... /*2m*/ to switch on mode 2

MODE 4: shutter-revolver No. 2, collimated Heusler mode (not fully implemented yet (no polarized neutron simulation is possible in McStas up to now) C1,C2,C3,C4=20,40,60 min, vertical focussing on monochromator (horizontal focus fix), no horizontal focussing on analyzer (vertical focus fix)

MODE 5: shutter-revolver No. 2, collimated Heusler mode (not fully implemented yet (no polarized neutron simulation is possible in McStas up to now) only C1,C2=20,40,60 min, vertical focussing on monochromator (horizontal focus fix), horizontal focussing on analyzer (vertical focus fix)

To change the modes of the spectrometer a program mode was written (see appendix A) in the programming language Perl, which is available on any standard linux/unix system. This is a kind of front-end and serves to prepare the McStas programs for the different configurations of the instrument [3].

In addition the monochromators, collimators, the sample and and the distance between sample and analyser (equal to distance analyser–detector 3) of the spectrometer can be changed using Perl programs (see appendix A).

3.3 Changing the Spectrometer Design

To optimize the operation of the spectrometer it is not only necessary to perform calculations in the different configurations of the spectrometer, but also to try different designs of the spectrometer.

One example for such a design feature is the outlay of the supermirror coatings of the neutron guide. Several types of calculation can be performed according to different supermirror coatings of the guide. The guide has three parts, part one is the first part after the source until the shutter. The second part is movable and can rotate, the geometry depending on the spectrometer mode. The third part is fix again, in this part the collimators C1 can be inserted. According to different types of supermirror coating the following guides have been implemented for simulation:

- all parts of the guide coated left,right, top and bottom with supermirrors (\input{guides1} in file panda.ins)
- all parts of the guide coated left and right with supermirrors, nowhere top and bottom supermirrors (\input{guides2} in file panda.ins)
- without any supermirrors in any part of the guide (\input{guides3} in file panda.ins)
- with top and bottom supermirrors only in inner fix part of guide, no top and bottom supermirrors in movable and outer part of the guide, left and right supermirrors in all parts of the guide - **this is how the guide is designed** (\input{guides4} in file panda.ins)
- with top and bottom supermirrors in inner fix part and in movable rotatable part of guide,left and right supermirrors in all parts of the guide (\input{guides5} in file panda.ins)

To change the guide of the spectrometer a program guide was written (see appendix A). This is another front end for the McStas programs and changes the type of guide used for the simulation.

3.4 Storing the Results of the Simulations

Calculations have been performed for several parameters, configurations and designs of the instrument.The results of the calculation are stored according to the following directory convention:

```
# the DIRECTORY consists of a name such as
# /li2_4/de0_5/guide2/m1/dia0_02/cu111/dsa1_2/Q1_3/Ef2_2
# meaning
# li2_4 ...initial wavelength in Angstroem(alternatively Ei1_2
#          if you want to specify energy in meV)
# de0_5 ....energy interval for source in meV (alternatively dl0.4
#          if you want to specify wavelength interval)
# guide2 .. guide type 2 (alternatively guide1 - guide5)
# m1 ...... mode 1 (alternatively m2/c1_60, m3/c1_40/c2_40/c3_40/c4_40
#           m4/c1_40/c2_40/c3_40/c4_40 (meaning mode 2
```

[3] the program mode just changes the comments /*m1*/ etc. in specified files

```
#           collimator1 divergence 60, mode 3 collimator C1 div 40, C2 div 40
#           etc, collimator C1 is  obligatory in mode 2-4, C2-C4 in mode 3-4,
#            in mode 5 only C1,C2))
# dia0_02...diaphragm width in m
# cu111.... copper monochromator (and analyzer) 111 reflection (alternatively pg002, hs)
# dsa1_2 .. distance sample analyzer in m
# Q1.3 .... scattering vector (in Angstroem^-1)
# Ef2.2.... final energy in meV (alternatively wavelength lf in angstroem)
```

3.5 Displaying results

The output of the simulation can be viewed via the `mcplot` and `mcdisplay` commands. In addition the (`Perl`) programs `mcplot,mcresplot` were modified into `mcplotmr,mcresplotmr` to enable automatic control and generation of a postscript file (see appendix A).

3.6 Automatic Control and Generation of this Report

To do several simulations in a sequence and create the figures and tables shown in this report the `Perl` programs `do_sim2` and `sim` have been written (see appendix A).

`sim` can be used to perform a simulation just by giving the directory according to the convention of section 3.4 as an argument. The results of the calculations are displayed, a line for a table is created in `Latex` format and appended to `table.tex`. In addition a postscript figure containing the plots for this document is created and a corresponding statement added to the file `figure.tex`.

The program `do_sim2` just calls `sim` subsequently with different arguments (i.e. simulation parameters-configurations-designs).

If a simulation run has been performed successfully the program `sim` will not recalculate the results unless the files are deleted before running another simulation. Instead only the plots will be generated from the data files. If `sim` should perform

1. a display of the instrument
2. ask the operator to choose some nice viewgraphs of the instrument and
3. add such figure to figures.tex

then it is just necessery to put a comment before the command "exit;" in the last part of program sim.

4 Results

In the following figures the main results of the calculation are assembled. From top left to right bottom they show: divergence-, position- and energy distribution on the analyzer [4],divergence-, position- and energy distribution at the position of the detector and (for focussing modes) behind the 15mm rectangular slit at the position of the detector. Note that the position distribution is plotted always "looking into the beam".

The tables show the results corresponding to the different figures.

[4] mind that the divergence distribution is always taken by a monitor (of fixed size) in front of the analyzer normal to the incoming beam, the position and energy distribution are calculated directly on the analyzer , i.e. depend on the scattering angle

Table 1: Flux Calculations with wavelength independent source flux of $4.5{\cdot}10^{12}\ \frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}$ corresponding to the different figures

source parameters				spectrometer details		spectral flux			total flux	simulation results
wave-length [Å]	energy [meV]	dλ [Å]	dE [meV]	dia-phragm1 [m]	figure number	Monochro-mator $\frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}$	5x5 cm^2 sample $\frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}$	1x1 cm^2 sample $\frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2{\cdot}\AA}}$	1x1 cm^2 sample $\frac{\mathrm{n}}{\mathrm{s{\cdot}cm^2}}$	Directory
6.2	2.1282	0.1457	0.05	0.04	2	9.30e+07	1.91e+08	3.94e+08	2.87e+07	/li6_2/de0_05/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_
5	3.2723	0.1223	0.08	0.04	3	8.42e+07	3.70e+08	7.89e+08	4.82e+07	/li5_0/de0_08/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_
4	5.1129	0.1566	0.2	0.04	4	7.82e+07	3.38e+08	7.71e+08	6.03e+07	/li4_0/de0_2/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_0
2.4	14.2026	0.1695	1	0.04	5	7.11e+07	3.40e+08	9.64e+08	8.14e+07	/li2_4/de1/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/
1.5	36.3587	0.2088	5	0.04	6	5.83e+07	2.27e+08	7.30e+08	7.53e+07	/li1_5/de5/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/
6.2	2.1282	0.1457	0.05	0.04	7	7.60e+07	1.32e+08	2.78e+08	2.02e+07	/li6_2/de0_05/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_
5	3.2723	0.1223	0.08	0.04	8	6.89e+07	2.54e+08	5.46e+08	3.34e+07	/li5_0/de0_08/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_
4	5.1129	0.1566	0.2	0.04	9	6.39e+07	2.28e+08	5.23e+08	4.09e+07	/li4_0/de0_2/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_0
2.4	14.2026	0.1695	1	0.04	10	5.83e+07	2.17e+08	6.14e+08	5.19e+07	/li2_4/de1/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/
1.5	36.3587	0.2088	5	0.04	11	5.03e+07	1.41e+08	4.94e+08	5.10e+07	/li1_5/de5/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/
6.2	2.1282	0.1457	0.05	0.04	12	4.41e+07	5.05e+07	1.14e+08	8.30e+06	/li6_2/de0_05/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_
5	3.2723	0.1223	0.08	0.04	13	4.00e+07	9.62e+07	2.22e+08	1.36e+07	/li5_0/de0_08/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_
4	5.1129	0.1566	0.2	0.04	14	3.71e+07	8.40e+07	2.08e+08	1.63e+07	/li4_0/de0_2/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_0
2.4	14.2026	0.1695	1	0.04	15	3.38e+07	7.55e+07	2.31e+08	1.95e+07	/li2_4/de1/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/
1.5	36.3587	0.2088	5	0.04	16	3.22e+07	4.82e+07	1.90e+08	1.96e+07	/li1_5/de5/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/
6.2	2.1282	0.1457	0.05	0.02	17	2.37e+08	9.59e+08	1.84e+09	1.34e+08	/li6_2/de0_05/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf
5	3.2723	0.1223	0.08	0.02	18	1.85e+08	1.53e+09	3.23e+09	1.97e+08	/li5_0/de0_08/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf
4	5.1129	0.1566	0.2	0.02	19	1.43e+08	1.20e+09	2.61e+09	2.04e+08	/li4_0/de0_2/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf4
2.4	14.2026	0.1695	1	0.02	20	9.39e+07	8.73e+08	2.18e+09	1.84e+08	/li2_4/de1/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf2_4
1.5	36.3587	0.2088	5	0.02	21	8.02e+07	6.30e+08	1.57e+09	1.62e+08	/li1_5/de5/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf1_5

Table 2: Flux Calculations with wavelength independent source flux of $4.5 \cdot 10^{12}$ $\frac{\mathrm{n}}{\mathrm{s \cdot cm^2 \cdot Å}}$ corresponding to the different figures

source parameters				spectrometer details		spectral flux			total flux	simulation results
wave-length [Å]	energy [meV]	dλ [Å]	dE [meV]	dia-phragm1 [m]	figure number	Monochro-mator $\frac{\mathrm{n}}{\mathrm{s \cdot cm^2 \cdot Å}}$	5x5 cm^2 sample $\frac{\mathrm{n}}{\mathrm{s \cdot cm^2 \cdot Å}}$	1x1 cm^2 sample $\frac{\mathrm{n}}{\mathrm{s \cdot cm^2 \cdot Å}}$	1x1 cm^2 sample $\frac{\mathrm{n}}{\mathrm{s \cdot cm^2}}$	Directory
6.2	2.1282	0.1457	0.05	0.04	22	9.30e+07	1.91e+08	3.94e+08	2.87e+07	/li6_2/de0_05/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/
5	3.2723	0.1223	0.08	0.04	23	8.42e+07	3.70e+08	7.91e+08	4.83e+07	/li5_0/de0_08/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/
4	5.1129	0.1566	0.2	0.04	24	7.82e+07	3.38e+08	7.71e+08	6.03e+07	/li4_0/de0_2/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/p
2.4	14.2026	0.1695	1	0.04	25	7.11e+07	3.40e+08	9.60e+08	8.11e+07	/li2_4/de1/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg0
1.5	36.3587	0.2088	5	0.04	26	5.84e+07	2.27e+08	7.33e+08	7.56e+07	/li1_5/de5/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg0
6.2	2.1282	0.1457	0.05	0.04	27	7.60e+07	1.32e+08	2.79e+08	2.03e+07	/li6_2/de0_05/guide5/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/

Table 3: Flux Calculations with wavelength dependent source flux corresponding to the different figures

source parameters				spectrometer details		spectral flux			total flux	simulation results
wave-length [Å]	energy [meV]	dλ [Å]	dE [meV]	dia-phragm1 [m]	figure number	Monochro-mator $\frac{n}{s \cdot cm^2 \cdot Å}$	5x5 cm^2 sample $\frac{n}{s \cdot cm^2 \cdot Å}$	1x1 cm^2 sample $\frac{n}{s \cdot cm^2 \cdot Å}$	1x1 cm^2 sample $\frac{n}{s \cdot cm^2}$	Directory
6.2	2.1282	0.1457	0.05	0.04	2	1.12e+07	2.29e+07	4.72e+07	3.44e+06	/li6_2/de0_05/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_
5	3.2723	0.1223	0.08	0.04	3	1.91e+07	8.38e+07	1.79e+08	1.09e+07	/li5_0/de0_08/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_
4	5.1129	0.1566	0.2	0.04	4	3.01e+07	1.30e+08	2.97e+08	2.32e+07	/li4_0/de0_2/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_0
2.4	14.2026	0.1695	1	0.04	5	6.39e+07	3.05e+08	8.66e+08	7.31e+07	/li2_4/de1/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/
1.5	36.3587	0.2088	5	0.04	6	8.44e+07	3.29e+08	1.06e+09	1.09e+08	/li1_5/de5/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/
6.2	2.1282	0.1457	0.05	0.04	7	9.11e+06	1.58e+07	3.33e+07	2.42e+06	/li6_2/de0_05/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_
5	3.2723	0.1223	0.08	0.04	8	1.56e+07	5.75e+07	1.24e+08	7.56e+06	/li5_0/de0_08/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_
4	5.1129	0.1566	0.2	0.04	9	2.46e+07	8.77e+07	2.01e+08	1.57e+07	/li4_0/de0_2/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_0
2.4	14.2026	0.1695	1	0.04	10	5.24e+07	1.95e+08	5.52e+08	4.66e+07	/li2_4/de1/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/
1.5	36.3587	0.2088	5	0.04	11	7.28e+07	2.04e+08	7.15e+08	7.38e+07	/li1_5/de5/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/
6.2	2.1282	0.1457	0.05	0.04	12	5.29e+06	6.05e+06	1.37e+07	9.95e+05	/li6_2/de0_05/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_
5	3.2723	0.1223	0.08	0.04	13	9.06e+06	2.18e+07	5.03e+07	3.08e+06	/li5_0/de0_08/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_
4	5.1129	0.1566	0.2	0.04	14	1.43e+07	3.23e+07	8.00e+07	6.27e+06	/li4_0/de0_2/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_0
2.4	14.2026	0.1695	1	0.04	15	3.04e+07	6.78e+07	2.08e+08	1.75e+07	/li2_4/de1/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/
1.5	36.3587	0.2088	5	0.04	16	4.66e+07	6.98e+07	2.75e+08	2.84e+07	/li1_5/de5/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/
6.2	2.1282	0.1457	0.05	0.02	17	2.84e+07	1.15e+08	2.21e+08	1.61e+07	/li6_2/de0_05/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf
5	3.2723	0.1223	0.08	0.02	18	4.19e+07	3.47e+08	7.32e+08	4.46e+07	/li5_0/de0_08/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf
4	5.1129	0.1566	0.2	0.02	19	5.50e+07	4.62e+08	1.00e+09	7.85e+07	/li4_0/de0_2/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf4
2.4	14.2026	0.1695	1	0.02	20	8.44e+07	7.84e+08	1.96e+09	1.65e+08	/li2_4/de1/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf2_4
1.5	36.3587	0.2088	5	0.02	21	1.16e+08	9.12e+08	2.27e+09	2.35e+08	/li1_5/de5/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf1_5

Table 4: Flux Calculations with wavelength dependent source flux corresponding to the different figures

source parameters				spectrometer details		spectral flux			total flux	simulation results
wave-length [Å]	energy [meV]	dλ [Å]	dE [meV]	dia-phragm1 [m]	figure number	Monochro-mator $\frac{\text{n}}{\text{s}\cdot\text{cm}^2\cdot\text{Å}}$	5x5 cm^2 sample $\frac{\text{n}}{\text{s}\cdot\text{cm}^2\cdot\text{Å}}$	1x1 cm^2 sample $\frac{\text{n}}{\text{s}\cdot\text{cm}^2\cdot\text{Å}}$	1x1 cm^2 sample $\frac{\text{n}}{\text{s}\cdot\text{cm}^2}$	Directory
6.2	2.1282	0.1457	0.05	0.04	22	1.12e+07	2.29e+07	4.72e+07	3.44e+06	/li6_2/de0_05/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/
5	3.2723	0.1223	0.08	0.04	23	1.91e+07	8.38e+07	1.79e+08	1.09e+07	/li5_0/de0_08/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/
4	5.1129	0.1566	0.2	0.04	24	3.01e+07	1.30e+08	2.97e+08	2.32e+07	/li4_0/de0_2/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/p
2.4	14.2026	0.1695	1	0.04	25	6.39e+07	3.05e+08	8.62e+08	7.29e+07	/li2_4/de1/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg0
1.5	36.3587	0.2088	5	0.04	26	8.45e+07	3.29e+08	1.06e+09	1.09e+08	/li1_5/de5/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg0
6.2	2.1282	0.1457	0.05	0.04	27	9.11e+06	1.58e+07	3.35e+07	2.43e+06	/li6_2/de0_05/guide5/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/

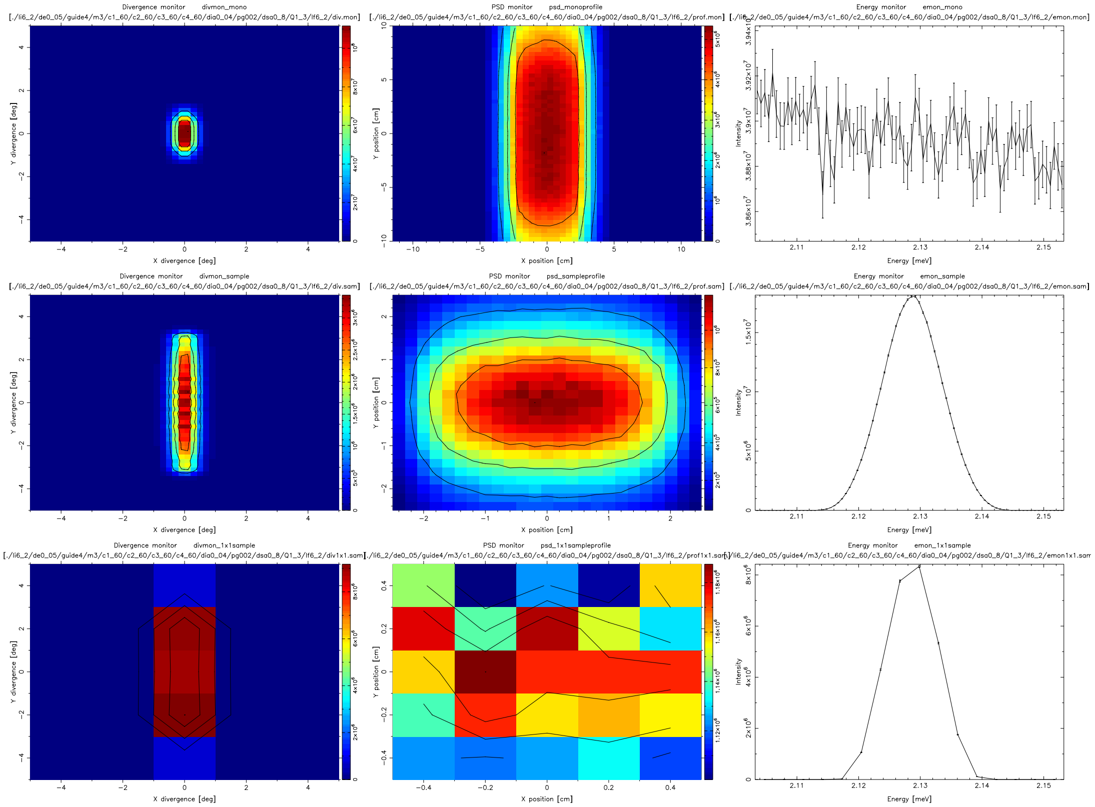

Figure 2: ../li6_2/de0_05/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf6_2/mcstas.ps

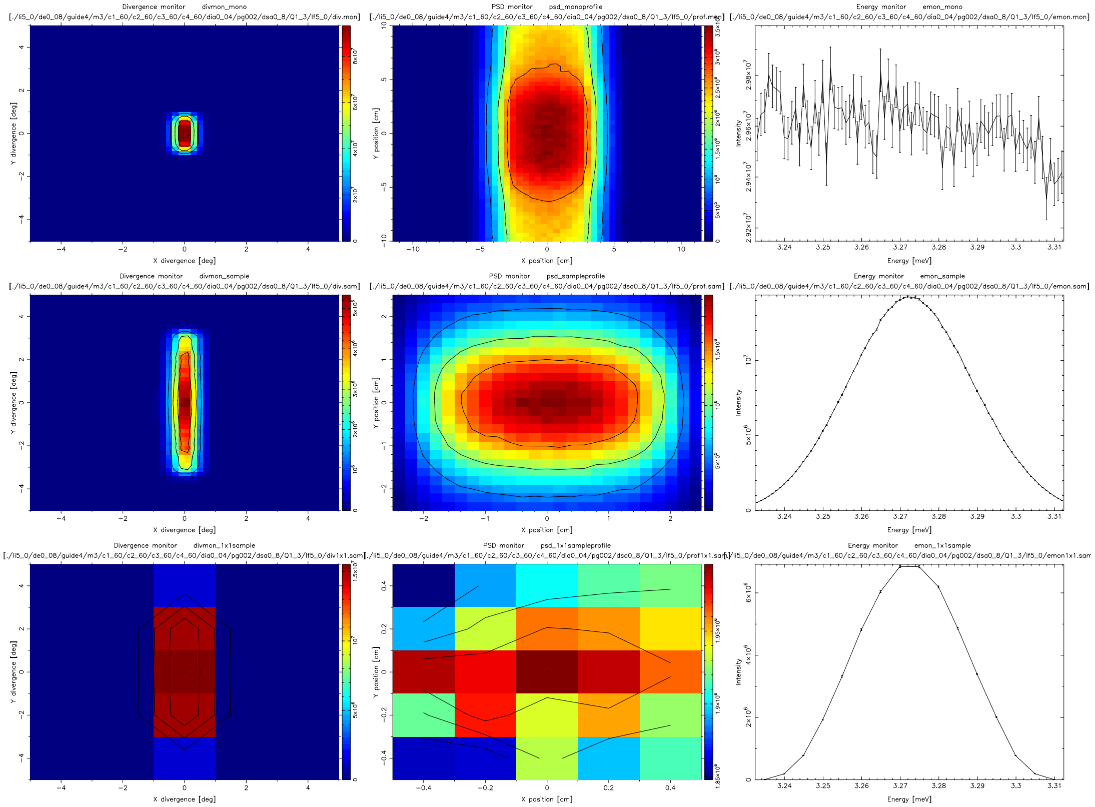

Figure 3: ../li5_0/de0_08/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf5_0/mcstas.ps

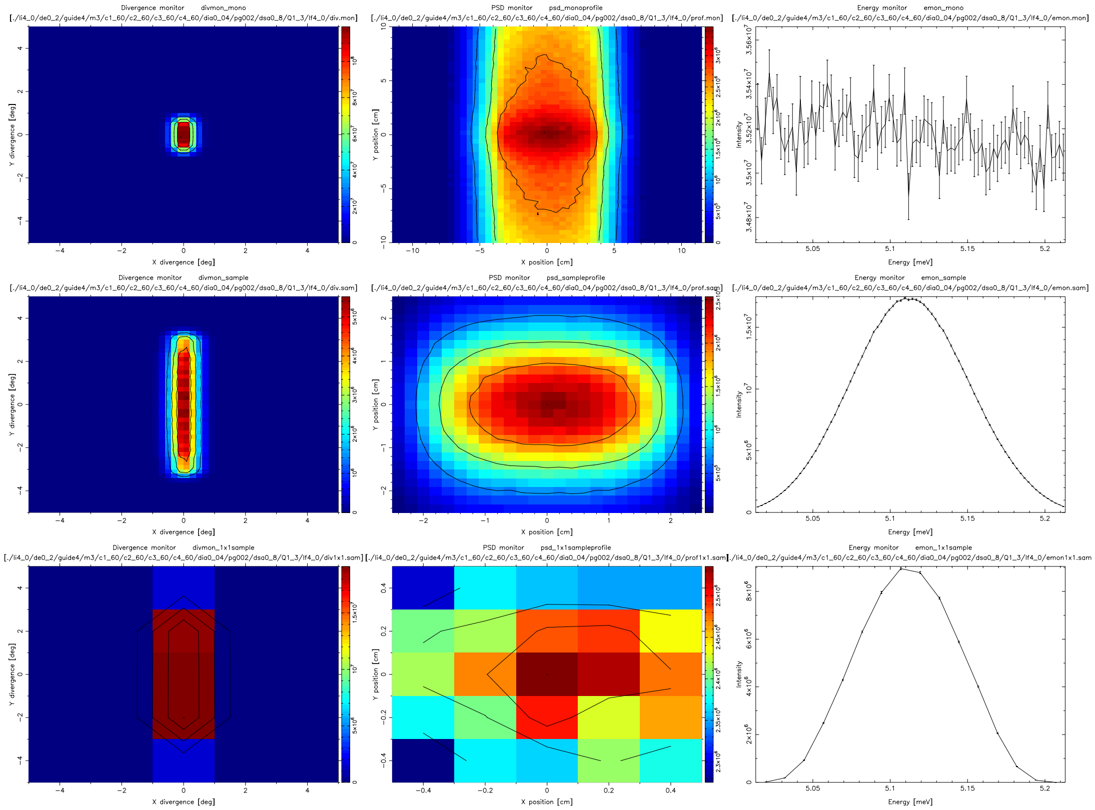

Figure 4: ../li4_0/de0_2/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf4_0/mcstas.ps

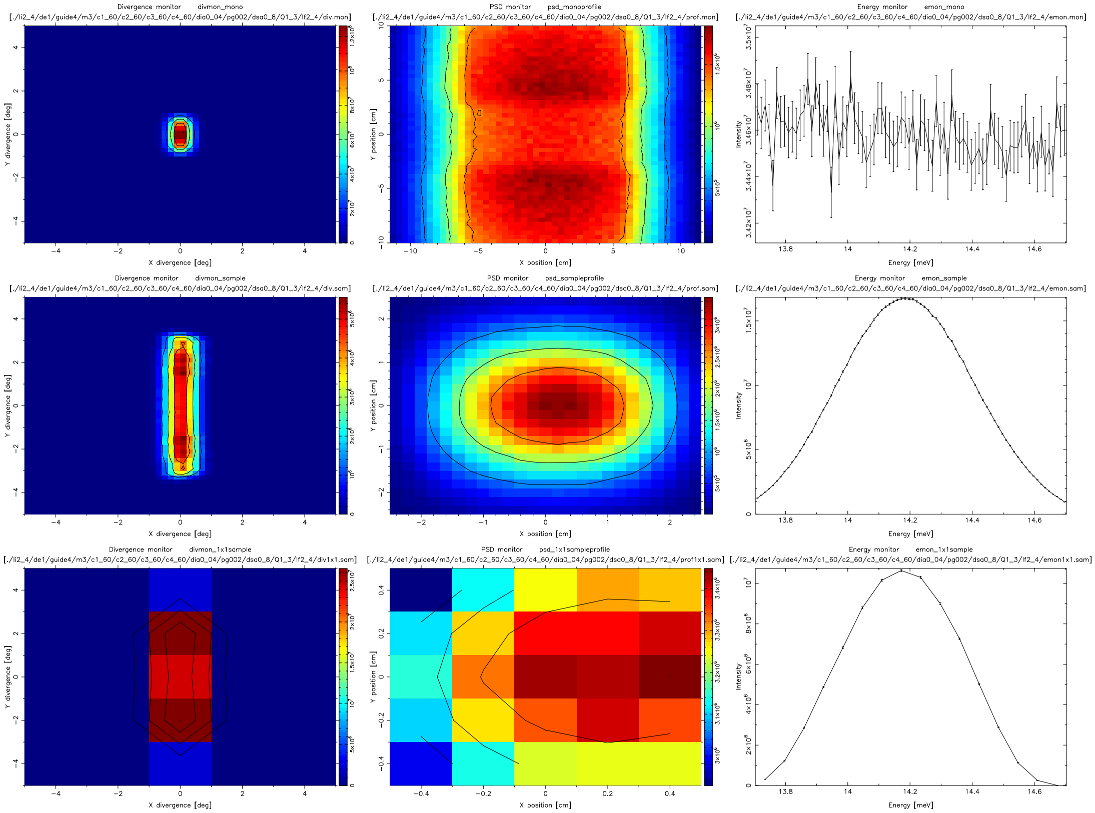

Figure 5: ../li2_4/de1/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf2_4/mcstas.ps

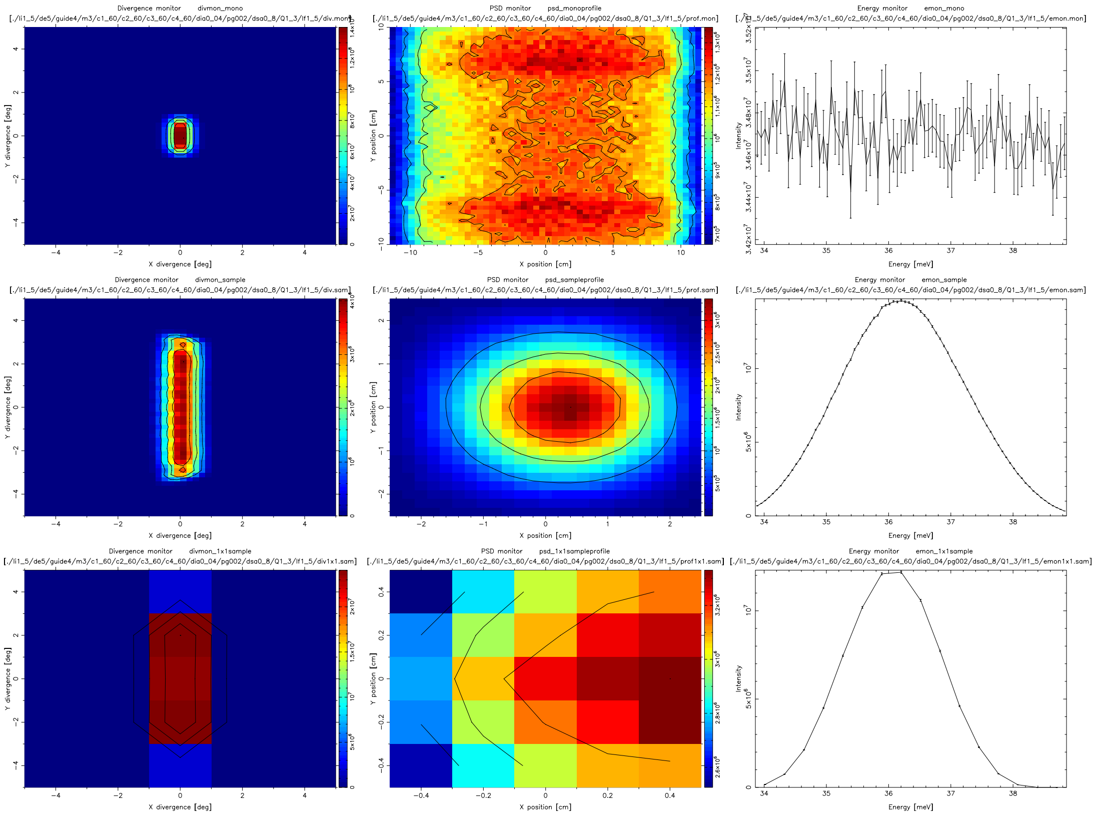

Figure 6: ../li1_5/de5/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf1_5/mcstas.ps

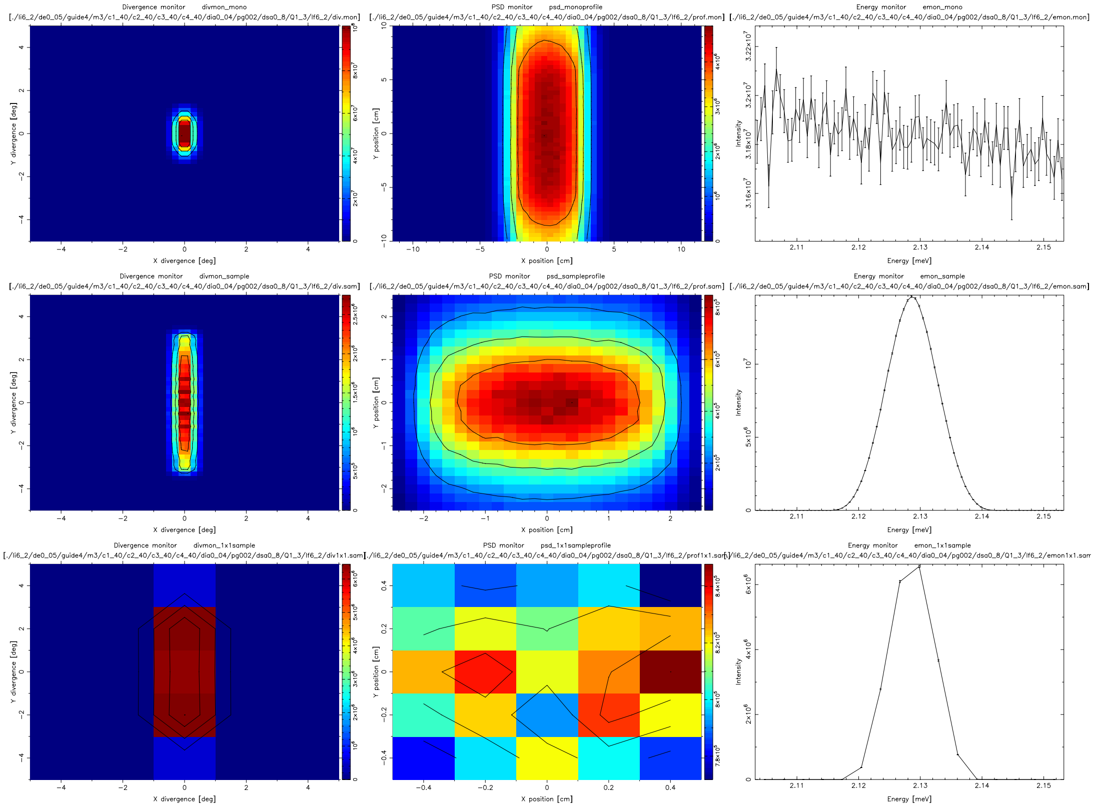

Figure 7: ../li6_2/de0_05/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/pg002/dsa0_8/Q1_3/lf6_2/mcstas.ps

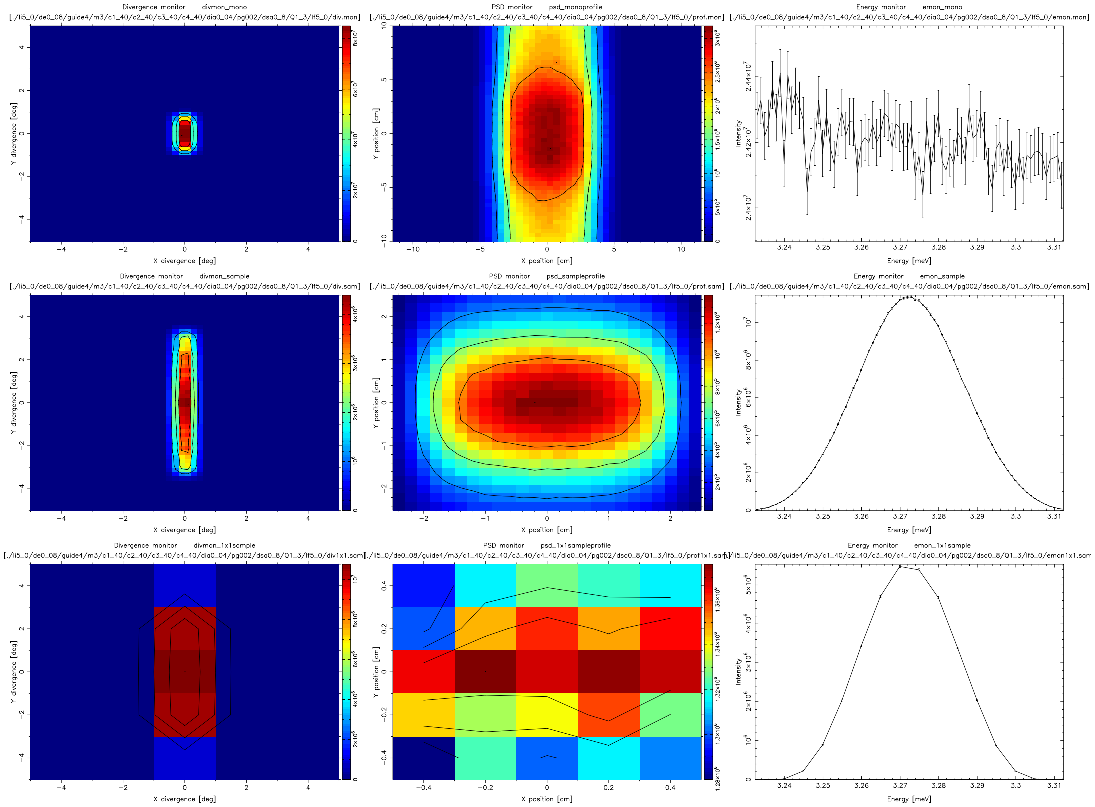

Figure 8: ../li5_0/de0_08/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/pg002/dsa0_8/Q1_3/lf5_0/mcstas.ps

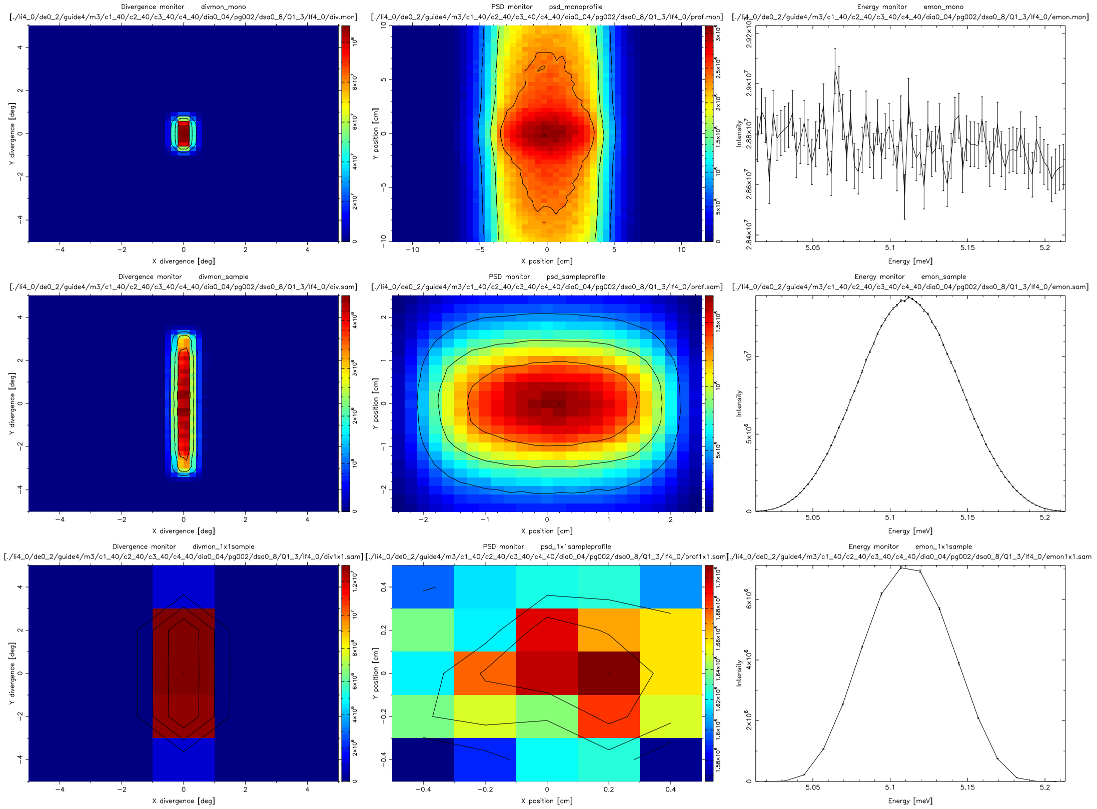

Figure 9: ../li4_0/de0_2/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/pg002/dsa0_8/Q1_3/lf4_0/mcstas.ps

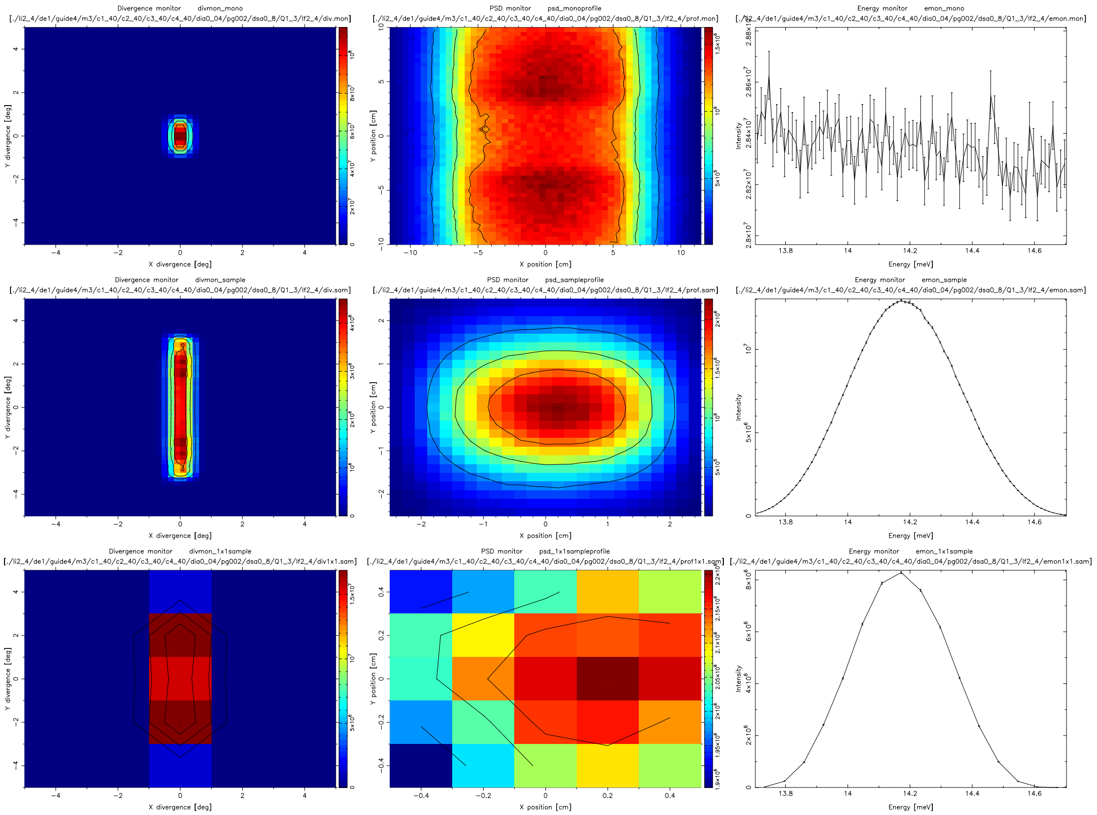

Figure 10: ../li2_4/de1/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/pg002/dsa0_8/Q1_3/lf2_4/mcstas.ps

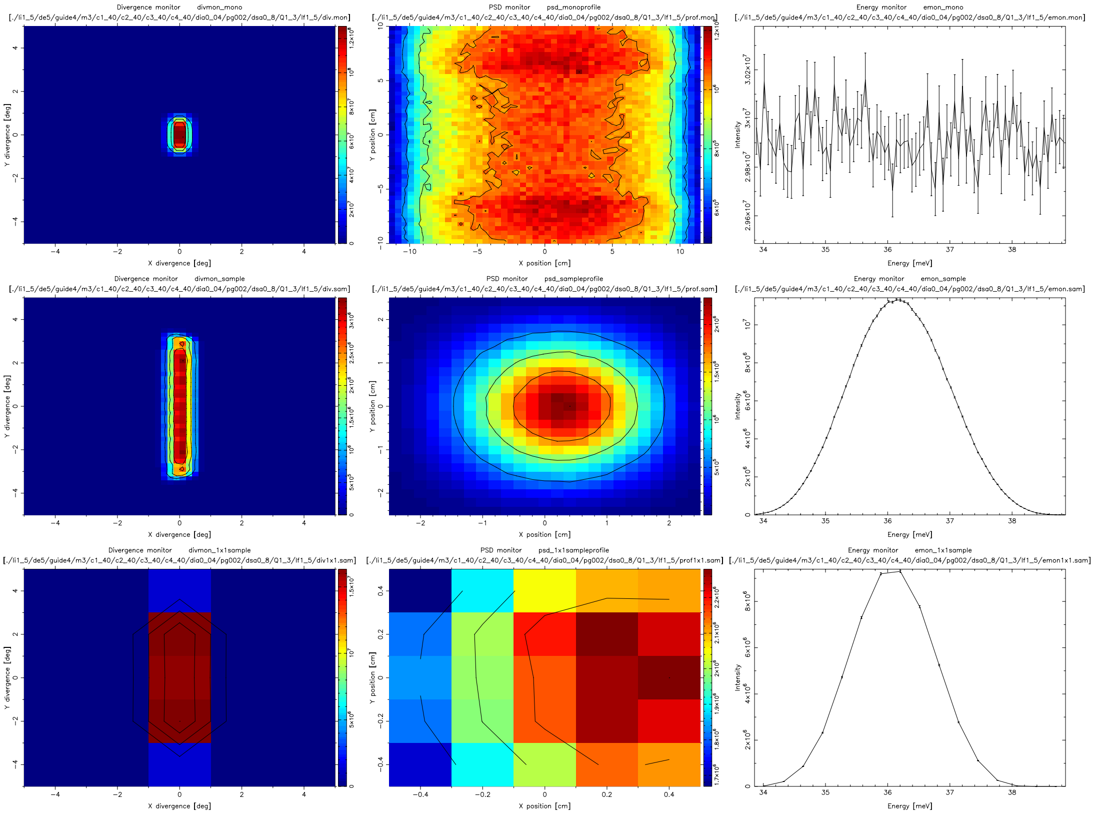

Figure 11: ../li1_5/de5/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/pg002/dsa0_8/Q1_3/lf1_5/mcstas.ps

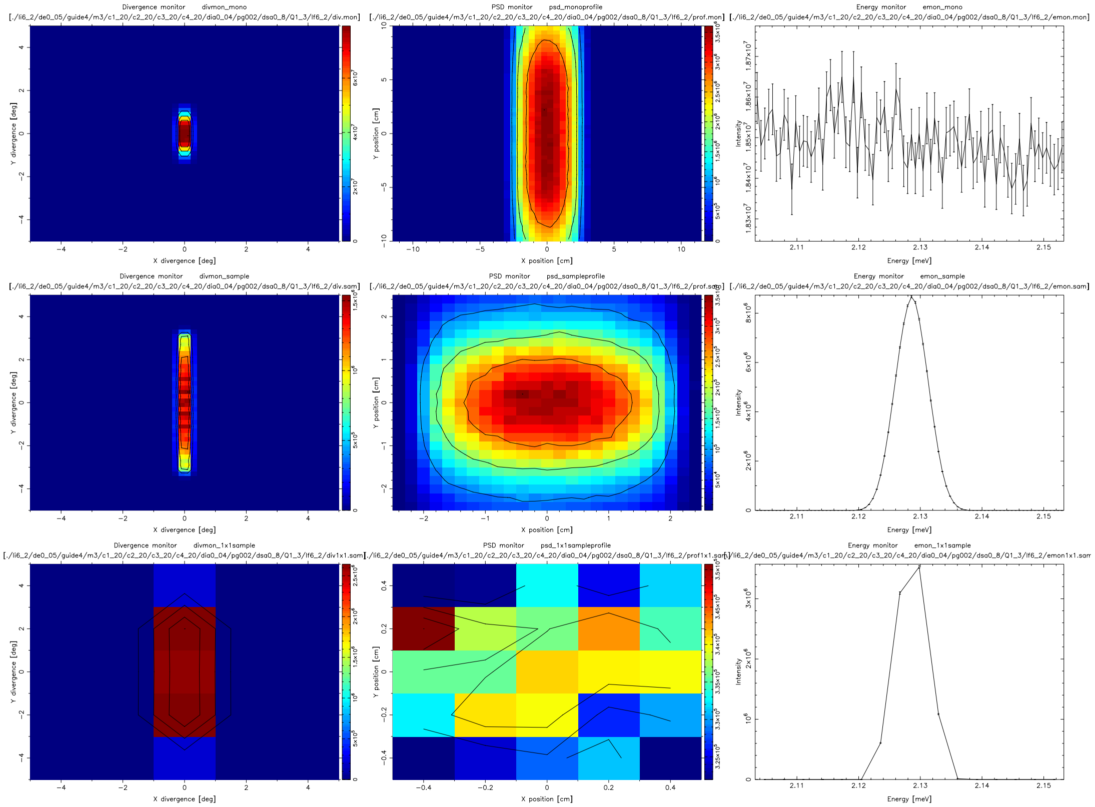

Figure 12: ../li6_2/de0_05/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/pg002/dsa0_8/Q1_3/lf6_2/mcstas.ps

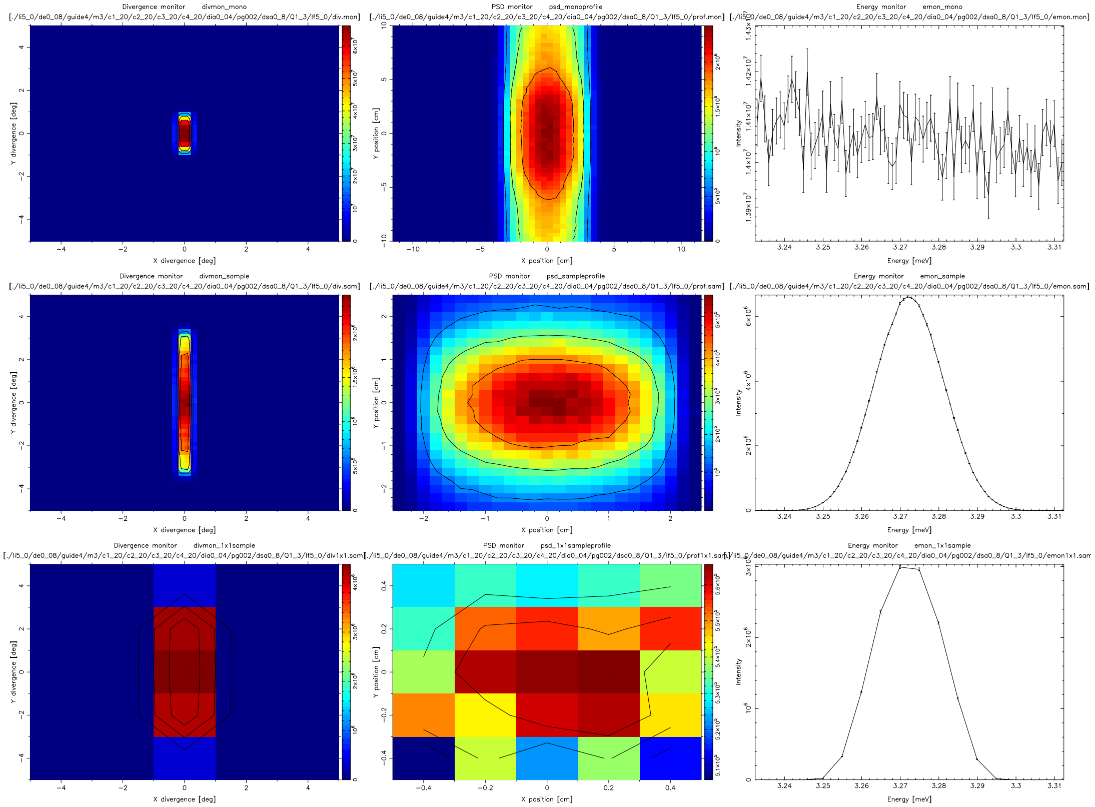

Figure 13: ../li5_0/de0_08/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/pg002/dsa0_8/Q1_3/lf5_0/mcstas.ps

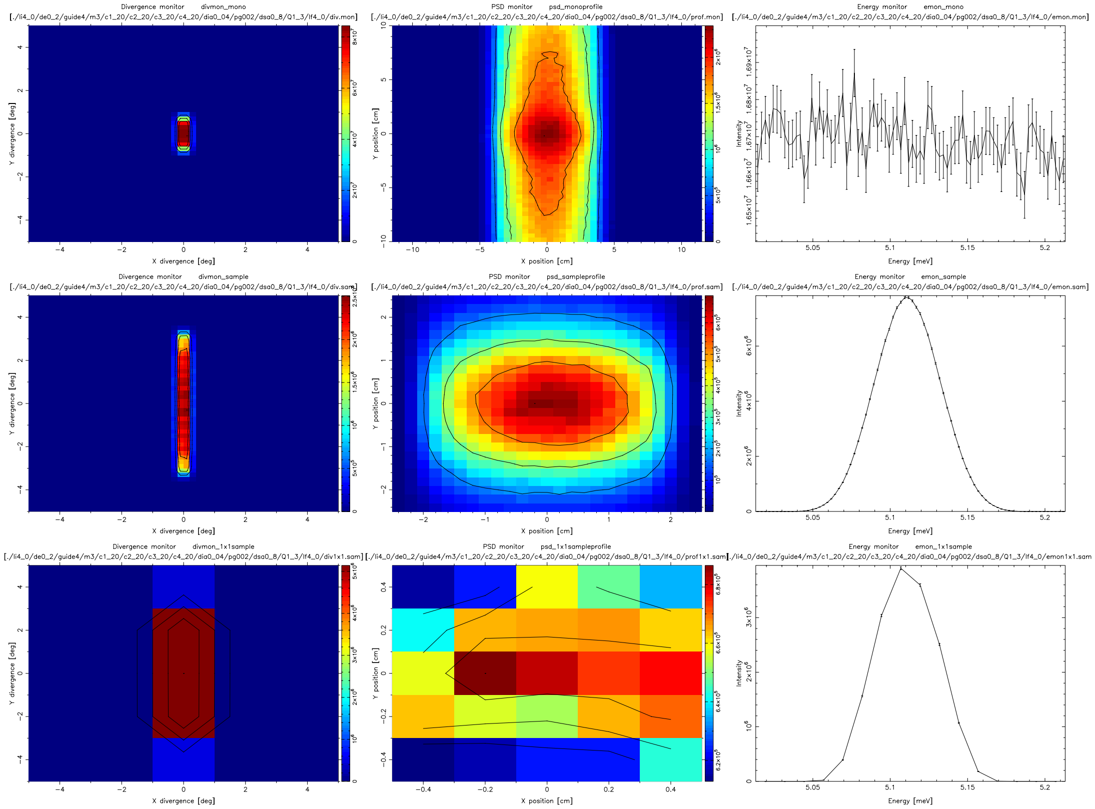

Figure 14: ../li4_0/de0_2/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/pg002/dsa0_8/Q1_3/lf4_0/mcstas.ps

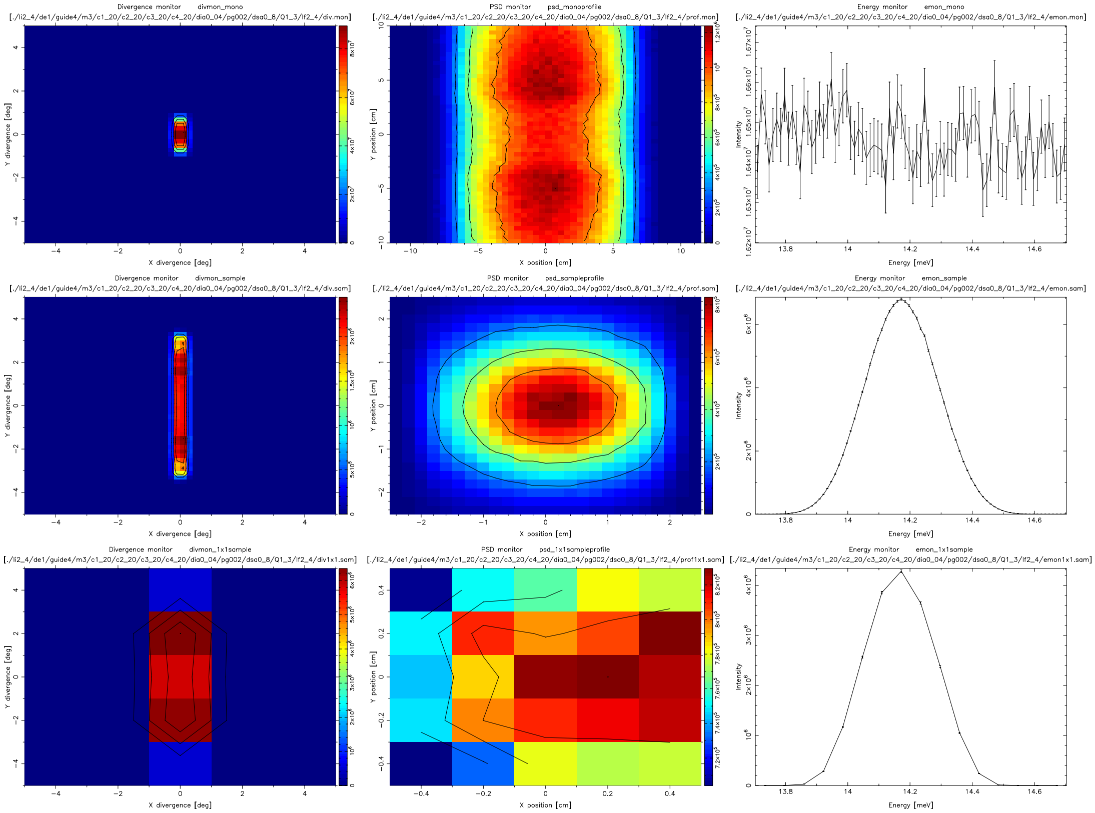

Figure 15: ../li2_4/de1/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/pg002/dsa0_8/Q1_3/lf2_4/mcstas.ps

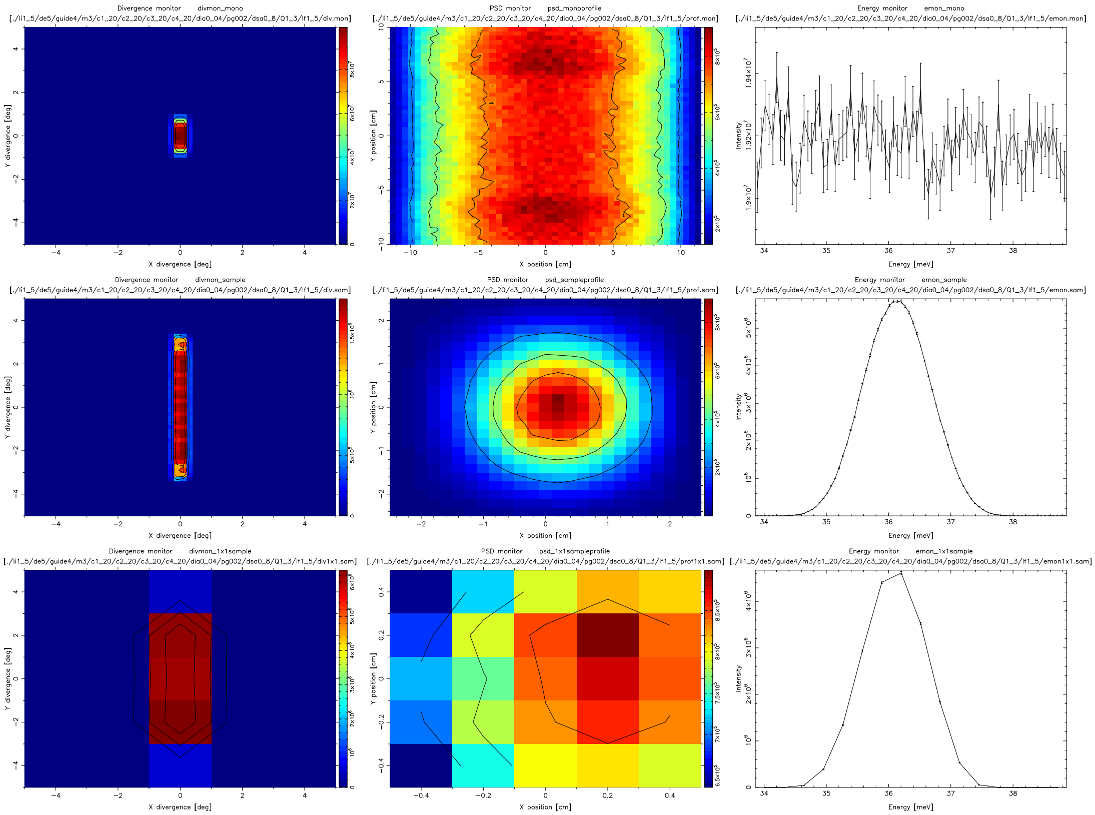

Figure 16: ../li1_5/de5/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/pg002/dsa0_8/Q1_3/lf1_5/mcstas.ps

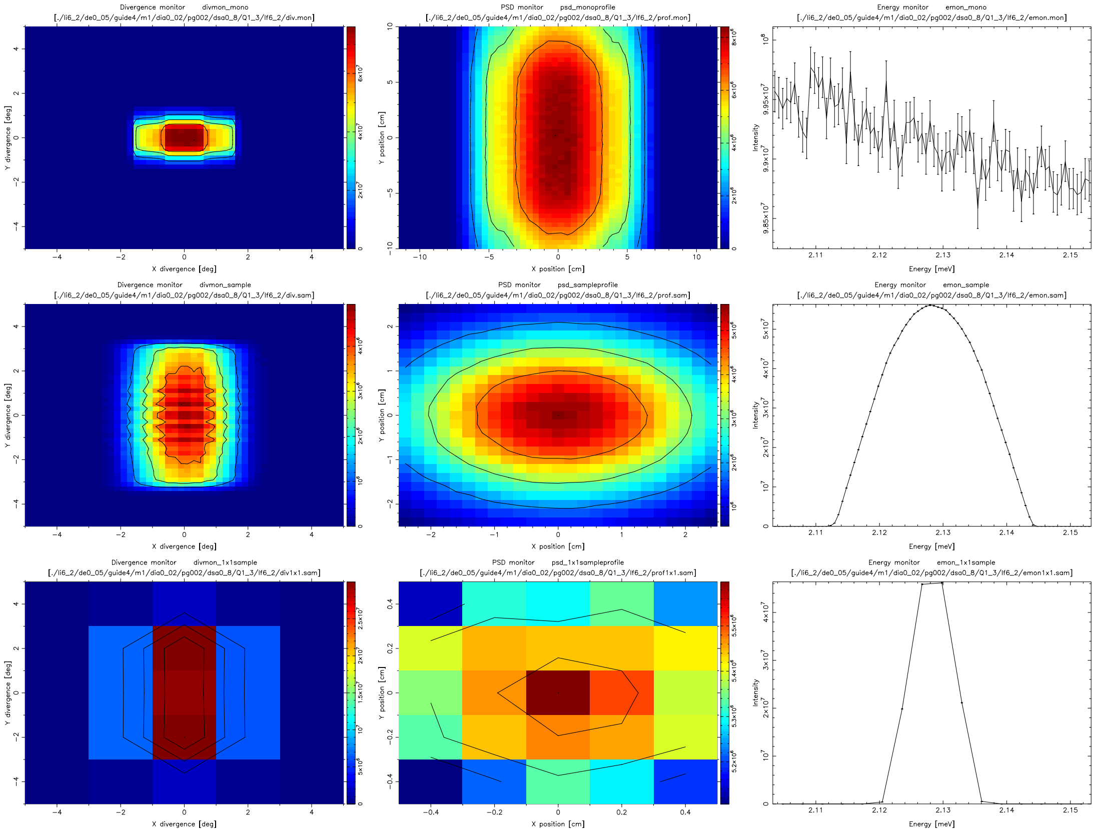

Figure 17: ../li6_2/de0_05/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf6_2/mcstas.ps

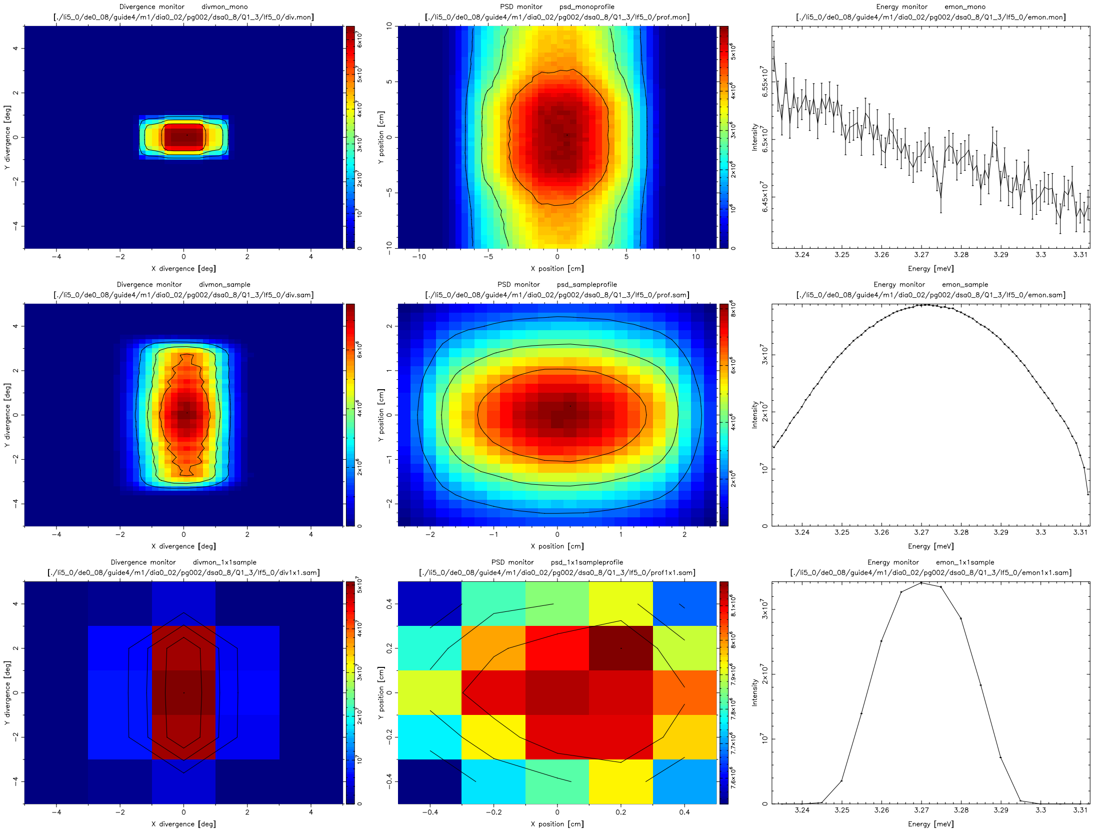

Figure 18: ../li5_0/de0_08/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf5_0/mcstas.ps

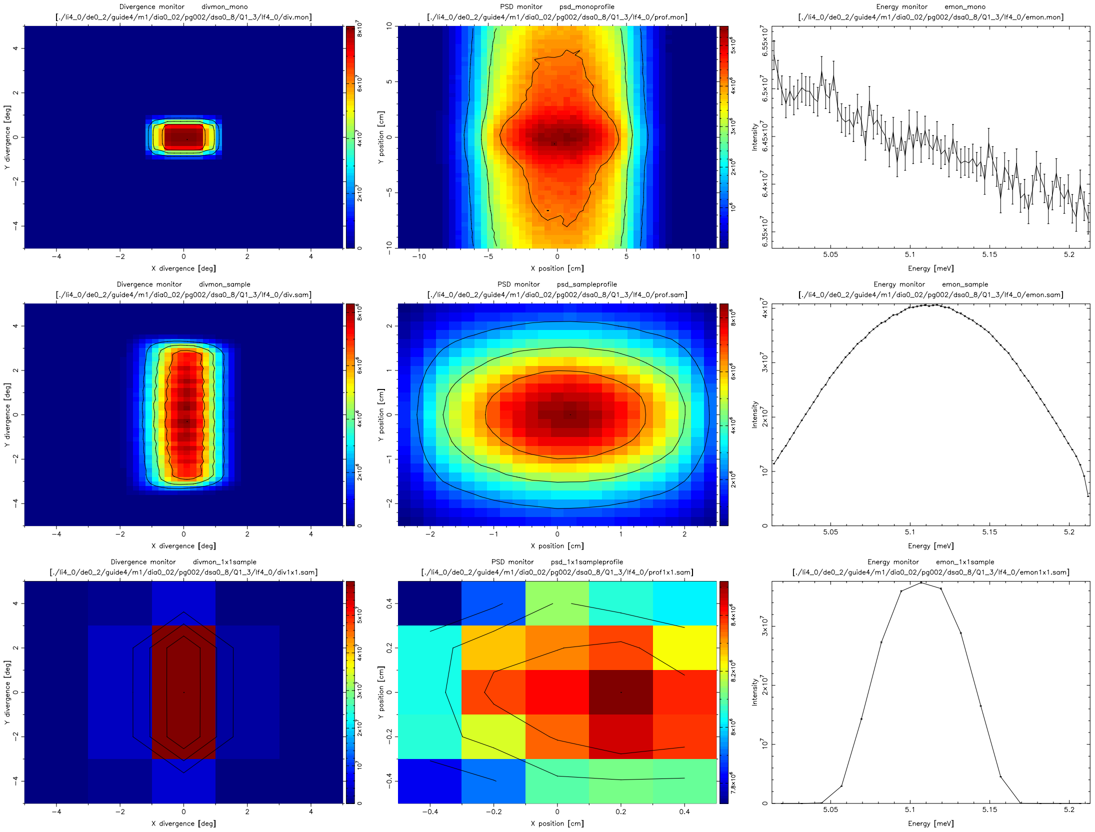

Figure 19: ../li4_0/de0_2/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf4_0/mcstas.ps

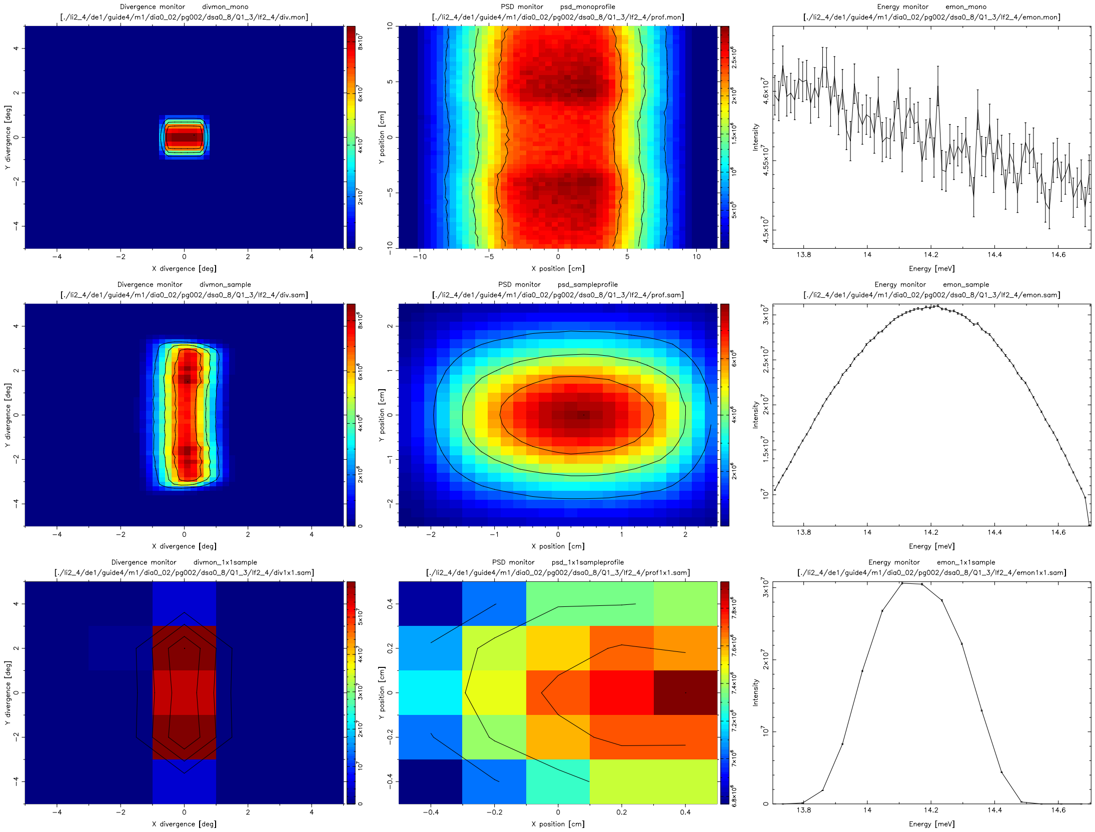

Figure 20: ../li2_4/de1/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf2_4/mcstas.ps

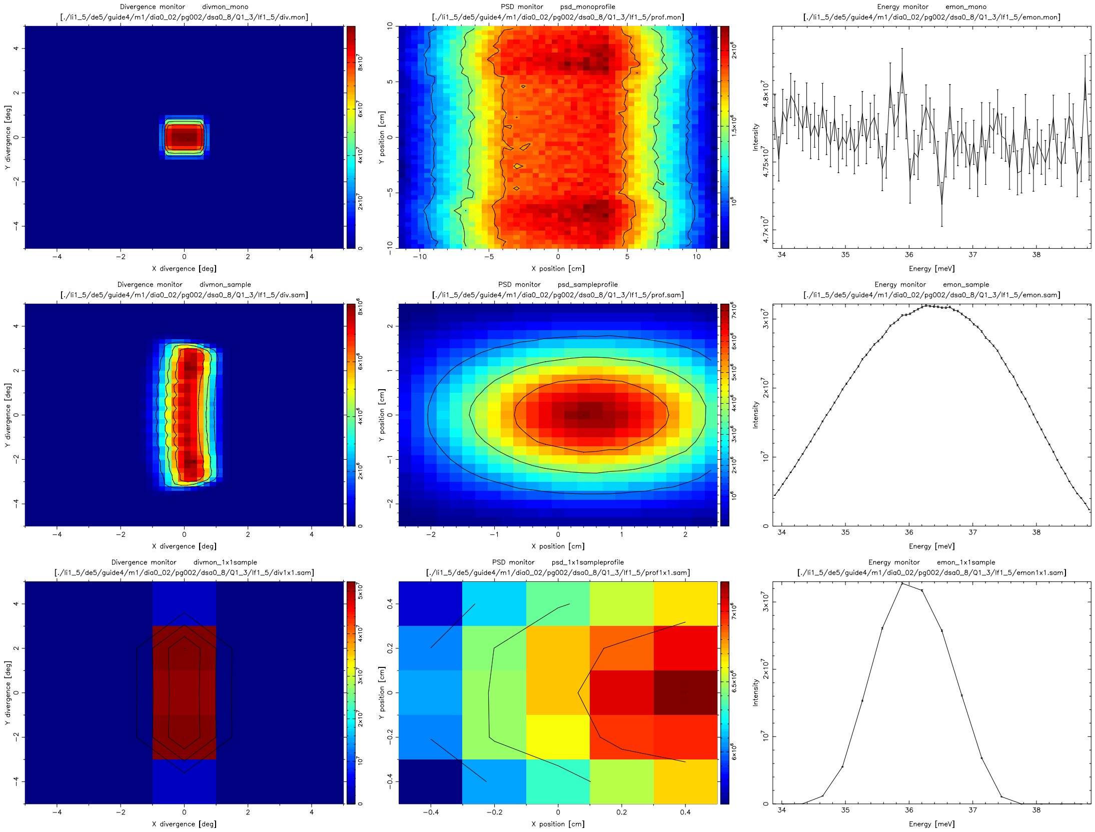

Figure 21: ../li1_5/de5/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3/lf1_5/mcstas.ps

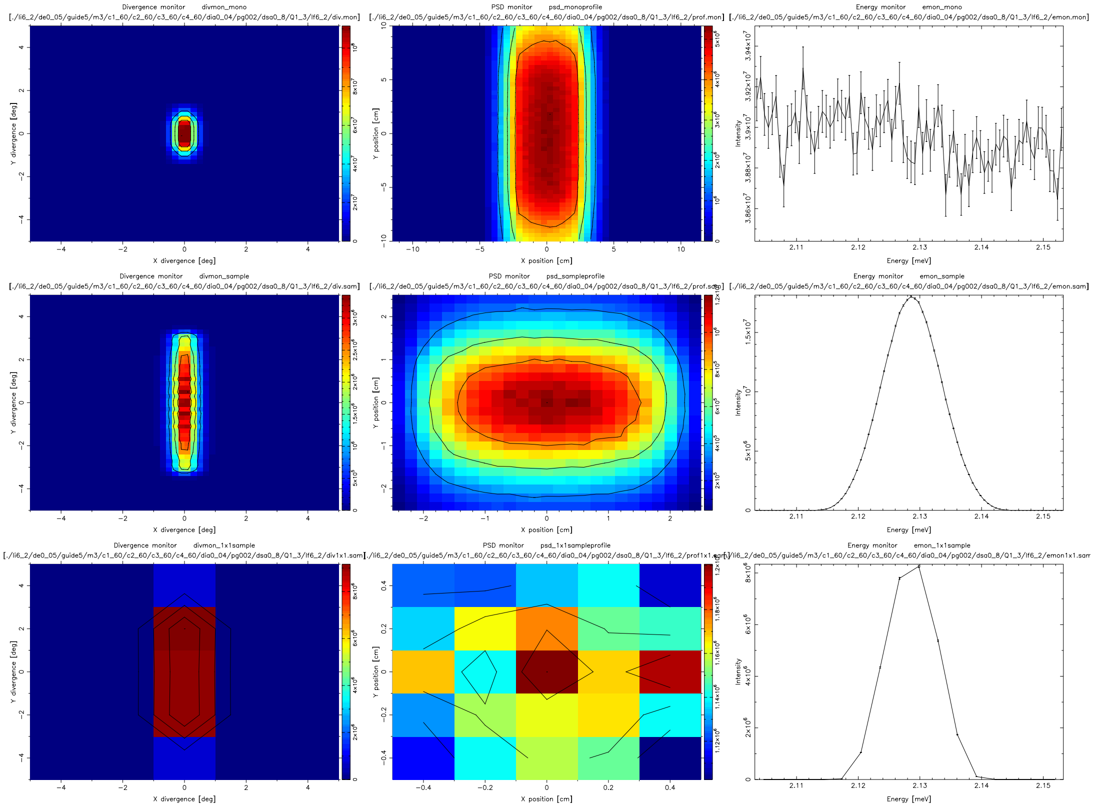

Figure 22: ../li6_2/de0_05/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf6_2/mcstas.ps

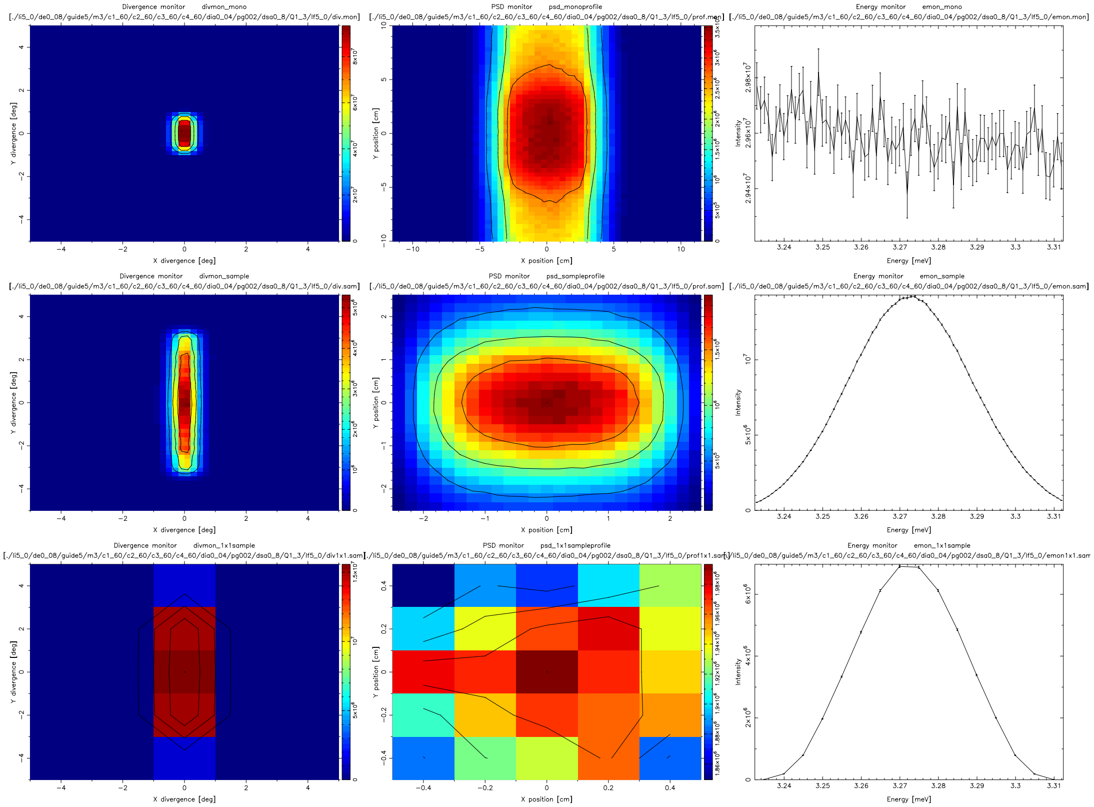

Figure 23: ../li5_0/de0_08/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf5_0/mcstas.ps

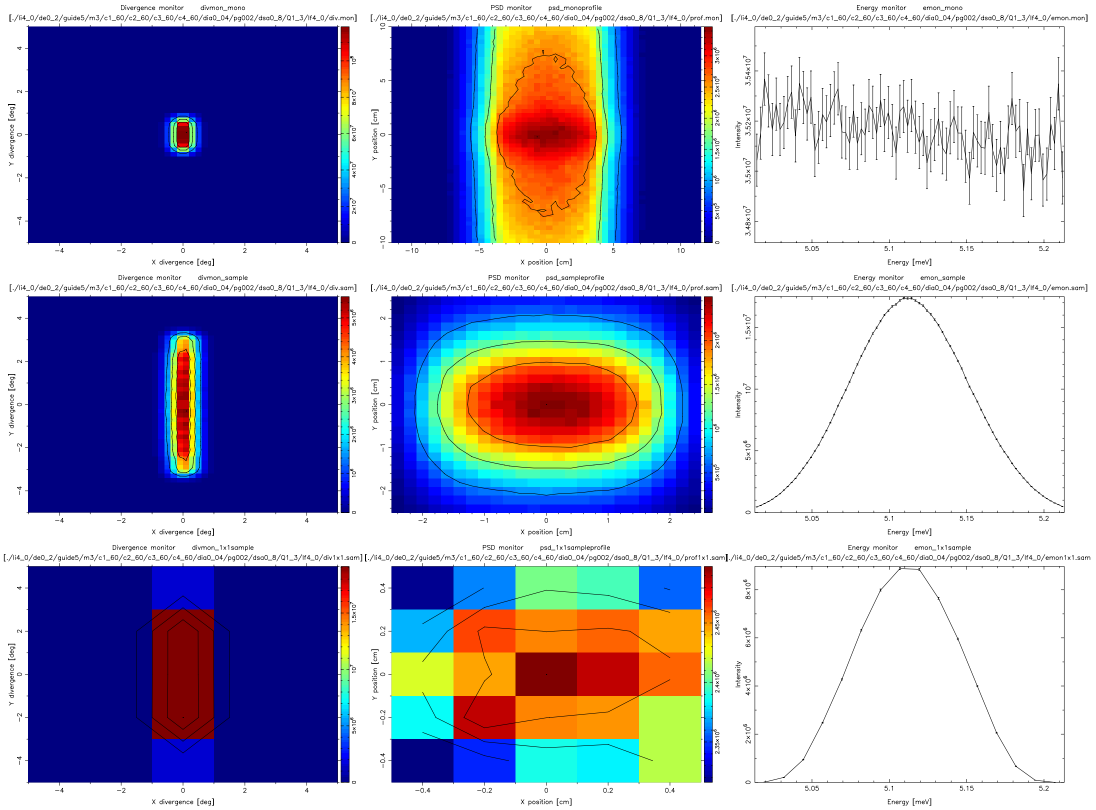

Figure 24: ../li4_0/de0_2/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf4_0/mcstas.ps

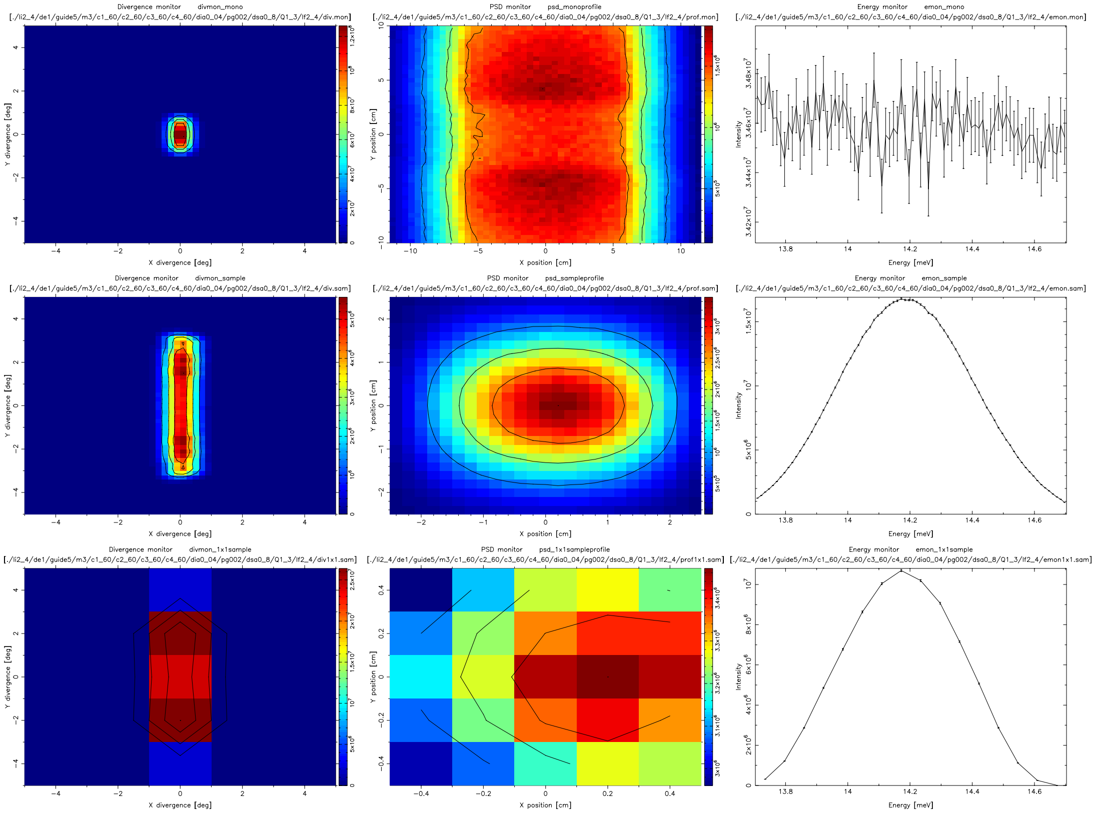

Figure 25: ../li2_4/de1/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf2_4/mcstas.ps

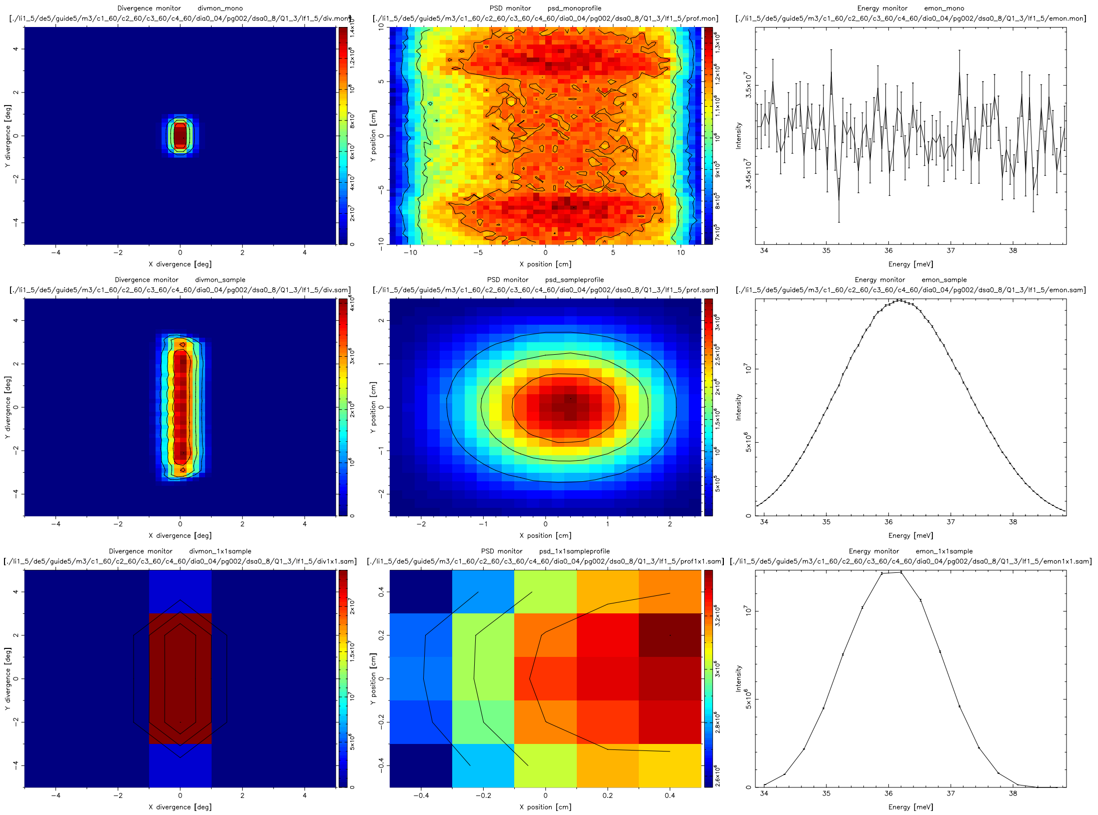

Figure 26: ../li1_5/de5/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3/lf1_5/mcstas.ps

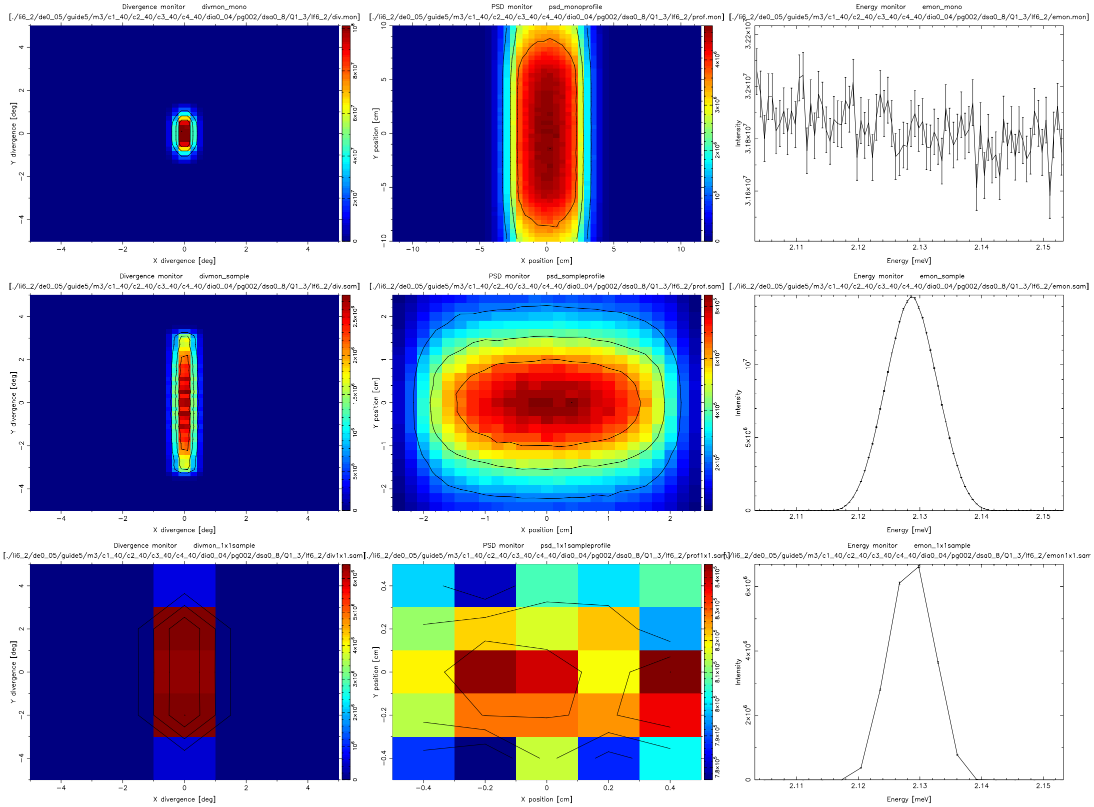

Figure 27: ../li6_2/de0_05/guide5/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/pg002/dsa0_8/Q1_3/lf6_2/mcstas.ps

A Perl Program files

A.1 DO_SIM1ANEW

!/usr/bin/perl

```
## script to do the PANDA SIMulation according to mail Niels Pyka
## use as: DO_SIM1ANEW
## 18.03.00


#graphit monochromator
# energies (final energy equal initial energy)
@lide = ('/li6_2/de0_05','/li5_0/de0_08','/li4_0/de0_2','/li2_4/de1',
     '/li1_5/de5');
# initialize .tex files
open(Fout,">./table.tex");print Fout "\n";close Fout;
open(Fout,">./tablecor.tex");print Fout "\n";close Fout;
open(Fout,">./figure.tex");print Fout "\n";close Fout;


foreach(@lide)
{$lides=$_;($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
system("./SIMNEW ".$lides."/guide4/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3".$lf);}
open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

foreach(@lide)
{$lides=$_;($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
system("./SIMNEW ".$lides."/guide4/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/pg002/dsa0_8/Q1_3".$lf);}
open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

foreach(@lide)
{$lides=$_;($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
system("./SIMNEW ".$lides."/guide4/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/pg002/dsa0_8/Q1_3".$lf);}
open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

foreach(@lide)
{$lides=$_;($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;

system("./SIMNEW ".$lides."/guide4/m1/dia0_02/pg002/dsa0_8/Q1_3".$lf);}
open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;


# take care of initializing new table.tex file
system ("mv table.tex table1.tex");
open(Fout,">./table.tex");print Fout "\n";close Fout;

# take care of initializing new table.tex file
system ("mv tablecor.tex table1cor.tex");
open(Fout,">./tablecor.tex");print Fout "\n";close Fout;
#IV

foreach(@lide)
{$lides=$_;($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
system("./SIMNEW ".$lides."/guide5/m3/c1_60/c2_60/c3_60/c4_60/dia0_04/pg002/dsa0_8/Q1_3".$lf);}
open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

foreach(@lide)
{$lides=$_;($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
system("./SIMNEW ".$lides."/guide5/m3/c1_40/c2_40/c3_40/c4_40/dia0_04/pg002/dsa0_8/Q1_3".$lf);}
open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

foreach(@lide)
{$lides=$_;($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
system("./SIMNEW ".$lides."/guide5/m3/c1_20/c2_20/c3_20/c4_20/dia0_04/pg002/dsa0_8/Q1_3".$lf);}
open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

foreach(@lide)
{$lides=$_;($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
system("./SIMNEW ".$lides."/guide5/m1/dia0_02/pg002/dsa0_8/Q1_3".$lf);}
open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;


# we are not calculating heusler modes
#system ("mv table.tex table2.tex");
#exit 0;

#V
#1-4
#foreach(@lide)
#{$lides=$_; ($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
#system("./SIM
#".$lides."/guide5/m4/c1_20/c2_20/c3_20/c4_20/dia0_04/HS/dsa0_8/Q1_3".$lf);}
#open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;
```

```
#foreach(@lide)
#{$lides=$_; ($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
#system("./SIM
#".$lides."/guide5/m4/c1_40/c2_40/c3_40/c4_40/dia0_04/HS/dsa0_8/Q1_3".$lf);}
#open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

#foreach(@lide)
#{$lides=$_;  ($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
#system("./SIM
#".$lides."/guide5/m4/c1_60/c2_60/c3_60/c4_60/dia0_04/HS/dsa0_8/Q1_3".$lf);}
#open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

#foreach(@lide)
#{$lides=$_;  ($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
#system("./SIM
#".$lides."/guide5/m4/c1_20/c2_20/c3_20/c4_20/dia0_04/HS/dsa1_2/Q1_3".$lf);}
#open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

#foreach(@lide)
#{$lides=$_; ($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
#system("./SIM
#".$lides."/guide5/m4/c1_40/c2_40/c3_40/c4_40/dia0_04/HS/dsa1_2/Q1_3".$lf);}
#open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

#foreach(@lide)
#{$lides=$_; ($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
#system("./SIM
#".$lides."/guide5/m4/c1_60/c2_60/c3_60/c4_60/dia0_04/HS/dsa1_2/Q1_3".$lf);}
#open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

#5
#foreach(@lide)
#{$lides=$_;  ($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
#system("./SIM ".$lides."/guide5/m5/c1_60/c2_40/dia0_04/HS/dsa1_2/Q1_3".$lf);}
#open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

#foreach(@lide)
#{$lides=$_;  ($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
#system("./SIM ".$lides."/guide5/m5/c1_60/c2_40/dia0_04/HS/dsa0_8/Q1_3".$lf);}
#open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;

#VI
#foreach(@lide)
#{$lides=$_;  ($lf)=($lides=~m|(/[^/]+)/.*|);$lf=~s!i!f!;
#system("./SIM ".$lides."/guide5/m2/c1_60/dia0_04/HS/dsa0_8/Q1_3".$lf);}
#open(Fout,">>./figure.tex");print Fout '\\clearpage'."\n";close Fout;





#system ("mv table.tex table2.tex");



#
```

A.2 SIMNEW

!/usr/bin/perl

```
# script to perform PANDA simulation
# use as: sim DIRECTORY
# the DIRECTORY consists of a name such as
# /li2_4/de0_5/guide2/m1/dia0_02/cu111/dsa1_2/Q1_3/Ef2_2
# meaning
# li2_4 ...initial wavelength in Angstroem(alternatively Ei1_2
#           if you want to specify energy in meV)
# de0_5 ....energy interval for source in meV (alternatively dl0.4
#           if you want to specify wavelength interval)
# dia0_02...diaphragm width in m
# guide2 .. guide type 2 (alternatively guide1 - guide5)
# m1 ...... mode 1 (alternatively m2/c1_40, m3/c1_40/c2_40/c3_40/c4_40
#            m4/c1_40/c2_40/c3_40/c4_40 (meaning mode 2
#           collimator1 divergence 40, mode 3 collimator C1 div 40, C2 div 40
#           etc, collimator C1 is  obligatory in mode 2-4, C2-C4 in mode 3-4,
#            in mode 5 only C1,C2))
# cu111.... copper monochromator 111 reflection (alternatively pg002)
# dsa1_2...distance sample-analyzer (1.2m)
# Q1.3 .... scattering vector (in Angstroem^-1)
# Ef2.2.... final energy in meV (alternatively wavelength lf in angstroem)

# after the simulation a line is appended to the file table.tex, and a
# figure is added to file figure.tex


$ncount="9e8";

# store directory name
$dir=$ARGV[0];
@path=split('/',$dir);


#make directory if it is not existing
$dd=".";shift @path;
foreach (@path)
 {$dd.="/".$_; if (system ("ls ".$dd)) {system("mkdir ".$dd);print "->creating directory\n";} }

#calculate initial energy ei and corresponding wavelength li from directory name
$_=$path[0];
if (m/ei.*/i) {$neg="";} else {$neg="-";}
($eistr)=($path[0]=~m!\D+(\w+)!);$eistr=~s!_!.!;$eistr=$neg.$eistr;
if ($eistr < 0) {$li=-$eistr;$ei=sprintf("%4.4f",(81.807/$li/$li));}else{$ei=$eistr;
                 $li=sprintf("%4.4f",sqrt(81.807/$ei));}


#calculate energy interval for source from directory name
 $_=$path[1];if (m!de.*!i) {$neg="";} else {$neg="-";}
 ($destr)=($path[1]=~m!\D+(\w+)!);$destr=~s!_!.!;$destr=$neg.$destr;
 if ($destr < 0)
      {$dl=-$destr;
       $de=sprintf("%4.4f",(81.807/($li-$dl/2)/($li-$dl/2)-81.807/($li+$dl/2)/($li+$dl/2)));}
  else
       {$de=$destr;$dl=sprintf("%4.4f",sqrt(81.807/($ei-$de))-sqrt(81.807/($ei+$de)));}
```

```
# set guide in McStas program files using program guide
 ($no)=($path[2]=~m!.*(\d)!);system("./GUIDE ".$no);

# set mode in McStas program files using program mode
 ($mode)=($path[3]=~m!m(\d).*!i);system("./MODE ".$mode." *.INS");


# set collimators, if necessary
# m1 - no coll
# m2/c1_40
# m3/c1_40/c2_40/c3_40/c4_40
# m4/c1_40/c2_40/c3_40/c4_40
# m5/C1_60/c2_40
# collimator C1 is  obligatory in mode 2-4, C2-C4 in mode 3-4))


 #set collimator 1 in McStas program files using program coll
  if ($mode eq "2" or $mode eq "3" or $mode eq "4"or $mode eq "5") {
   ($no)=($path[4]=~m!c(\d)!i);
   ($div)=($path[4]=~m!_(\d+)!i);
   system("./COLL ".$no." ".$div." *.INS");
   shift @path;
   }

 #set collimator 2 in McStas program files using program coll
  if ($mode eq "3" or $mode eq "4"or $mode eq "5") {
   ($no)=($path[4]=~m!c(\d)!i);
   ($div)=($path[4]=~m!_(\d+)!i);
   system("./COLL ".$no." ".$div." *.INS");
   shift @path;
   }


 #set collimator 3 in McStas program files using program coll
  if ($mode eq "3" or $mode eq "4") {
   ($no)=($path[4]=~m!c(\d)!i);
   ($div)=($path[4]=~m!_(\d+)!i);
   system("./COLL ".$no." ".$div." *.INS");
   shift @path;
   }

 #set collimator 4 in McStas program files using program coll
  if ($mode eq "3" or $mode eq "4") {
   ($no)=($path[4]=~m!c(\d)!i);
   ($div)=($path[4]=~m!_(\d+)!i);
   system("./COLL ".$no." ".$div." *.INS");
   shift @path;
   }

#calculate  diaphragm width from directory name
($dia)=($path[4]=~m!\D+(\w+)!);$dia=~s!_!.!;
#set monochromator using program mon according to directory name
system("./MON".$path[5]);
#set distance sample - analyzer
($dsa)=($path[6]=~m!\D+(\w+)!);$dsa=~s!_!.!;
#set scattering vector Q as given by directory name
($q)=($path[7]=~m!\D+(\w+)!);$q=~s!_!.!;
```

```
#set final energy ef as given by directory name
$_=$path[8];if (m!ef.*!i) {$neg="";} else {$neg="-";}
($efstr)=($path[8]=~m!\D+(\w+)!);$efstr=~s!_!.!;$efstr=$neg.$efstr;
if ($efstr < 0) {$lf=-$efstr;$ef=sprintf("%4.4f",(81.807/$lf/$lf));}else{$ef=$efstr;
                 $lf=sprintf("%4.4f",sqrt(81.807/$ef));}

#----------------------------------------------------------------------------------------------
#if you do not want calculation just comment out the following few lines
# !!! do calculation only if there is no old calculation !!!!
if (-e  ".".$dir."/monitors.out")
{}
else
{
#print "Vanadium Sample\n";
#system "./SAMPLE V";
 print "mcstas compiling program (PANDA.INS)...\n";
 system "mcstas PANDA.INS";
 print "cc compiling program ...\n";
 system "cc -O -o panda PANDA.INS.c -lm";
     # delete directory of calculation - this is necessary for correct operation
     # of McStas
 system (  "rm  -r .".$dir);
 print "performing simulation ...\n";
       $string=" --dir=.".$dir." DIAPH1=".$dia." NEGLiPOSEi=".$eistr." NEGLfPOSEf=".$ef;
       $string.=" DNEGLPOSE=".$destr." Q=".$q." > ./monitors.out";print $string."\n";
  system ("./panda --ncount=".$ncount." ".$string);
  system ("mv ./monitors.out .".$dir);
# print "RESOLUTION SAMPLE\n";
# system "./SAMPLE R";
# system "mcstas PANDA.INS";
# print "cc compiling program ...\n";
# system "cc -O -o panda PANDA.INS.c -lm";
# print "performing simulation ...\n";
#       $string=" DIAPH1=".$dia." NEGLiPOSEi=".$eistr." NEGLfPOSEf=".$ef;
#       $string.=" DNEGLPOSE=".$destr." Q=".$q." > ./monitors.out";
##because of memory problems we can do only a ncount of 1e7 with ressample
#  system ("./panda --ncount=1e7 ".$string);
## system ("mv ./resmon.ell .".$dir);
#  system ("mv ./resmon.inel .".$dir);
}

#-------------------------------------------------------------------------

#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#extract table
 open(Fin,"<.".$dir."/monitors.out");
 @lines=<Fin>;$line= join (’’,@lines);
 close Fin;
#monochromator flux (n/scm2AA)
($fluxmon)=sprintf("%2.2e",($line=~m!.*\Qfluxmon_mono_I=\E(.+)\s\Qfluxmon_mono_ERR\E.*!));
#5x5cm sample flux (n/scm2AA)
($flux5x5spl)=sprintf("%2.2e",($line=~m!.*\Qfluxmon_sample_I=\E(.+)\s\Qfluxmon_sample_ERR\E.*!));
#1x1sample - flux (n/scm2AA)
($flu1x1spl)=  sprintf("%2.2e",($line=~m!.*\Qfluxmon_1x1sample_I=\E(.+)\s\Qfluxmon_1x1sample_ERR\E.*!));
#1x1sample - total flux (n/scm2)
$totflu1x1spl=sprintf("%2.2e",$flu1x1spl*$de/2/$ei*$li);
#detector - flux(n/scm2AA)
```

```
($detflux)=  sprintf("%2.3e",($line=~m!.*\Qfluxmon_detektor_I=\E(.+)\s\Qfluxmon_detektor_ERR\E.*!));
#detector - total flux(n/scm2)
$totdetflux=sprintf("%2.2e",$detflux*$de/2/$ei*$li);


#print table
# for latex substitute _ with \_ in $dir
$dirtex=$dir;
$dirtex=~s!\Q_\E!\\_!g;

  open(Fout,">>./table.tex");
  print Fout $li.’ & ’.$ei.’ & ’;
  print Fout $dl.’ & ’.$de.’ & ’.$dia.’ & ’;

# pictures in part1
  print Fout ’\ref{’.$dir.’/mcstas} & ’;
# pictures in part2
#  print Fout ’ &  \ref{’.$dir.’/mcstas}, \ref{’.$dir.’/pgplot} & ’;

# for monitor/samplefluxtable only in part1
      print Fout $fluxmon.’ & ’.$flux5x5spl.’ & ’.$flu1x1spl.’ & ’.$totflu1x1spl;
# for detector flux table only in part2
      #print Fout $detflux.’ & ’.$totdetflux;

print Fout ’ & ’.$dirtex.’  \\\\’."\n";
close Fout;

  open(Fout,">>./tablecor.tex");
  print Fout $li.’ & ’.$ei.’ & ’;
  print Fout $dl.’ & ’.$de.’ & ’.$dia.’ & ’;

$STRINGCOR=exp(30.3-0.53*$li)/4.5e12;

# pictures in part1
  print Fout ’\ref{’.$dir.’/mcstas} & ’;
# pictures in part2
#  print Fout ’ &  \ref{’.$dir.’/mcstas}, \ref{’.$dir.’/pgplot} & ’;

# for monitor/samplefluxtable only in part1
      print Fout sprintf("%2.2e",$fluxmon*$STRINGCOR).’ & ’. sprintf("%2.2e",$flux5x5spl*$STRINGCOR).’ &
# for detector flux table only in part2
      #print Fout $detflux.’ & ’.$totdetflux;

print Fout ’ & ’.$dirtex.’  \\\\’."\n";
close Fout;
#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#

####part1

system ("./MCPLOTMR.PL .".$dir."/mcstas.sim");
#move file mcstas.ps to correct directory
system ("mv ./mcstas.ps .".$dir);

####end part1

####part2
```

```
## make a plot using a modified version of mcplot.pl --- namely mcplotmr.pl
## this program is the same as mcplot.pl but automatically prints postscript
## file mcstas.ps and ends ...
## put mcstas.sim to part2.sim and modify so that only those detectors
## are plotted, which we want....
#   open (Fin, ".".$dir."/mcstas.sim");open (Fout, ">.".$dir."/part2.sim");
#    # read text from file into $line
#    @lines=<Fin>;$line= join ('',@lines);
#    # modify text in $line (as in program COLL)
#$line=~s!(begin data)[^']+('srcadapt).*?(end data)!\n!is;
#$line=~s!(begin data)[^b]+(res_detektor).*?(end data)!\n!is;
# # comment: s means treat $line as single line, i.e. match also newline by .
# # comment: .*? --- the question mark means that .* is "greedy" i.e. the first
# #          occcurence of "divergence" after coll  is taken
# #          (if ? is missing you get the
# #          last occurence in $line as match and this is not right)
# # output modified text
#   print Fout $line;   close Fin;   close Fout;
#system ("./MCPLOTMR.PL .".$dir."/part2.sim");
##move file mcstas.ps to correct directory
#system ("mv ./mcstas.ps .".$dir);
##system ("ps2epsi .".$dir."/mcstas.ps");
##system ("mv ./mcstas.epsi .".$dir);

####end part2

#extract figures and add to figure file
  open(Fout,">>./figure.tex");
   print Fout "\n".'\\begin{figure}[btp]%h=here, t=top, b=bottom, p=separate figure page'."\n";
   print Fout '\\begin{center}\\leavevmode'."\n";
#   print Fout '\\includegraphics [angle=-90,width=1.0\\textwidth]{/home/lero/Desktop/mcstas'.$dir.'/mcs
   print Fout '\\includegraphics [angle=-90,width=0.60\\textheight]{..'.$dir.'/mcstas.ps}'."\n";
  print Fout '\\caption{..'.$dirtex.'/mcstas.ps}'."\n".'\\label{'.$dir.'/mcstas}'."\n";
  print Fout '\\end{center}\\end{figure}'."\n\n";
close Fout;

#system ("./MCRESPLOTMR.PL .".$dir."/resmon.inel");
##move file pgplot.ps to correct directory
#system ("mv ./pgplot.ps .".$dir);
#
##extract figures and add to figure file
#  open(Fout,">>./figure.tex");
#   print Fout "\n".'\\begin{figure}[btp]%h=here, t=top, b=bottom, p=separate
#figure page'."\n";    print Fout '\\begin{center}\\leavevmode'."\n";
#   print Fout '\\includegraphics[angle=-90,width=1.0\\textwidth]{../'.$dir.               '/pgplot.ps}
#   print Fout
#'\\caption{'.$dirtex.'/pgplot.ps}'."\n".'\\label{'.$dir.'/pgplot}'."\n";
#print Fout '\\end{center}\\end{figure}'."\n\n";   close Fout;


#sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss

# comment 'exit' if do not want to be asked for plots of the spectrometer setup
 exit;

$i=0;
```

```
print "do you want a figure of".$dir."(Y/N)?";
if ( <STDIN>=~/^\s*y/i)
{
#-------------------------------------------------------------------------
print "compile program for tracing modus ... \n";
 system "mcstas --trace PANDA.INS";system "cc -O -o panda PANDA.INS.c -lm";
      # do display and ask for plot - generate file setup$i.ps
       $string=" DIAPH1=".$dia." NEGLiPOSEi=".$eistr." NEGLfPOSEf=".$ef;
       $string.=" DNEGLPOSE=".$destr." Q=".$q;print $string."\n";
#-------------------------------------------------------------------------
}
else
{exit;}

$i=0;
print "choose nice instrument view with middle mouse button and press Q to store\n";

do
{$i=$i+1;

  system ("./MCDISPMR.PL  --multi ./panda  ".$string);
  system ("cp ./mcstas.ps .".$dir."/setup".$i.".ps");


#add setup$i  to figure.tex - the figure file of report.tex
  open(Fout,">>./figure.tex");
   print Fout "\n".’\\begin{figure}[btp]%h=here, t=top, b=bottom, p=separate figure page
’."\n";
   print Fout ’\\begin{center}\\leavevmode’."\n";
   print Fout ’\\includegraphics[angle=0, width=1.0\\textwidth]{../’.$dir.’/setup’.$i.’.ps}’."\n";
   print Fout ’\\caption{’.$dirtex.’/setup’.$i.’.ps}’."\n".’\\label{’.$dir.’/setup’.$i.’}’."\n";
   print Fout ’\\end{center}\\end{figure}’."\n\n";
  close Fout;
  print "do you want another figure of".$dir."(Y/N)?";
}  while ( <STDIN>=~/^\s*y/i);

#ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
#
```

A.3 GUIDE

!/usr/bin/perl

```
unless ($ARGV[0] > 0)
{print " program guide used to change guide of panda mcstas simulation\n";
 print " usage: guide 1   -   sets program file panda.ins to guide 1 \n";
 exit 0;}

$guide=$ARGV[0];
if ($guide > 0 && $guide < 6)
  { $file="PANDA.INS";
   print "<".$file;
   open (Fin, $file);
   open (Fout, ">guide.out");
   while($line=<Fin>)
  {
```

```
  #switch off all guides
  $line=~s!(\Q%include\E\s*\Q"GUIDES\E)(\d)(\Q.INS"\E)\s!/*%include "GUIDES$2.INS"*/!i;
  #switch on specific guide
  $line=~
  s!\s*\Q/*\E.*\Q%include\E\s*\Q"GUIDES\E($guide)\Q.INS"\E\s*\Q*/\E!  %include "GUIDES$1.INS" !i;
  print Fout $line;
  }
   close Fin;
   close Fout;
   unless (rename "guide.out",$file)
     {die "\n error switching guide \n (perl rename cannot cross filesystems) \n";}
   print ">\t";
 print "\n panda guide ".$guide." switched on\n";
 }
else
 {die  "guide ".$guide." not supported \n";}
fi

#
```

A.4 MODE

!/usr/bin/perl

```
unless ($#ARGV > 0)
{print " program mode used to change mode of panda mcstas simulation\n";
 print " usage: mode 1 *.ins  sets program file *.ins to mode 1 \n";
 exit 0;}

$mode=$ARGV[0];
if ($mode > 0 && $mode < 6)
 {shift;
  foreach (@ARGV)
  {
   $file=$_;
   print "<".$file;
   open (Fin, $file);
   open (Fout, ">mode.out");
   while($line=<Fin>)
    {
    #switch off all modes
    $line=~s!\Q/*m1*/\E!/*m1  !g;
    $line=~s!\Q/*1m*/\E!  1m*/!g;
    $line=~s!\Q/*m2*/\E!/*m2  !g;
    $line=~s!\Q/*2m*/\E!  2m*/!g;
    $line=~s!\Q/*m3*/\E!/*m3  !g;
    $line=~s!\Q/*3m*/\E!  3m*/!g;
    $line=~s!\Q/*m4*/\E!/*m4  !g;
    $line=~s!\Q/*4m*/\E!  4m*/!g;
    $line=~s!\Q/*m5*/\E!/*m5  !g;
    $line=~s!\Q/*5m*/\E!  5m*/!g;
    #switch on specific mode
    if ($mode eq "1"){$line=~s!\Q/*m1\E\s+!/*m1*/!g;$line=~s!\s+\Q1m*/\E!/*1m*/!g;}
    if ($mode eq "2"){$line=~s!\Q/*m2\E\s+!/*m2*/!g;$line=~s!\s+\Q2m*/\E!/*2m*/!g;}
    if ($mode eq "3"){$line=~s!\Q/*m3\E\s+!/*m3*/!g;$line=~s!\s+\Q3m*/\E!/*3m*/!g;}
    if ($mode eq "4"){$line=~s!\Q/*m4\E\s+!/*m4*/!g;$line=~s!\s+\Q4m*/\E!/*4m*/!g;}
    if ($mode eq "5"){$line=~s!\Q/*m5\E\s+!/*m5*/!g;$line=~s!\s+\Q5m*/\E!/*5m*/!g;}
    print Fout $line;
    }
   close Fin;
   close Fout;
   unless (rename "mode.out",$file)
   {print "\n error switching mode \n (perl rename cannot cross filesystems) \n";exit 0;}
   print ">\t";}
 print "\n panda mode ".$mode." switched on\n";
 }
else
 {print "mode ".$mode." not supported \n";exit 0;}
fi

#
```

A.5 COLL

!/usr/bin/perl

```
unless ($#ARGV > 0)
{print " program coll used to change coll of panda mcstas simulation\n";
 print " usage: coll 1 40 *.ins  sets collimator 1 in program file *.ins to 40 arc minutes \n";
 print " the transmission is selected according to values calculated by\n";
 print " H. Schneider and depends on collimator type (1,2-4) and divergence \n";
 exit 0;}

$coll=$ARGV[0];

if ($coll > 0 && $coll < 5)
 {shift;
  $div=$ARGV[0];
  shift;
  $trans=1.0;
# set transmission value according to H. Schneiders calculation
  if ($coll eq "1")
   {if ($div eq "20") {$trans=0.85;}
    if ($div eq "40") {$trans=0.91;}
    if ($div eq "60") {$trans=0.95;}
   }
   else
   {if ($div eq "20") {$trans=0.945;}
    if ($div eq "40") {$trans=0.97;}
    if ($div eq "60") {$trans=0.98;}
   }
if ($trans eq 1.0) {print "ERROR PROGRAM COLL: Collimator ".$div." does not
exit\n"; exit 1;}

  foreach (@ARGV)
  {$file=$_;
   print "<".$file;
   open (Fin, $file);
   open (Fout, ">coll.out");

# read text from file into $line
    @lines=<Fin>;
    $line= join (’’,@lines);
    # modify text in $line
$line=~s!(COMPONENT\s*coll)($coll)(.*?divergence\s*=\s*){1}\d\d!$1$2$3$div!gis;
$line=~s!(COMPONENT\s*coll)($coll)(.*?transmission\s*=\s*){1}[\d\.]+!$1$2$3$trans!gis;
# comment: s means treat $line as single line, i.e. match also newline by .
# comment: .*? --- the question mark means that .* is "greedy" i.e. the first
#          occcurence of "divergence" after coll  is taken
#          (if ? is missing you get the
#          last occurence in $line as match and this is not right)

# output modified text
   print Fout $line;
   close Fin;
   close Fout;
   unless (rename "coll.out",$file)
   {print "\n error setting coll divergence \n (perl rename cannot cross filesystems) \n";exit 1;}
   print ">\t";}
 print "\n panda coll ".$coll." divergence set to ".$div." arc minutes \n";
 print "transmission = ".$trans."\n";
 }
else
 {print "coll ".$coll." not supported \n";exit 0;}
fi

#
```

A.6 MONPG002

!/usr/bin/perl

```
# program cu111 used to insert graphit monochromator  and analyzer
# into panda mcstas simulation
# usage: monpg002 (sets program file panda.ins to PG002 monochromator
# and analyzer)

   $file="PANDA.INS";
   print "<".$file;
   open (Fin, $file);
   open (Fout, ">mode.out");
   while($line=<Fin>)
    {
    #change mono parameters
    $line=~s!\QQ_mono=\E.*;!Q_mono= 1.8727;!;
    $line=~s!\Qr0_mono=\E.*;!r0_mono= 0.87;!;
    #put in copper monochromator and analyzer
    $line=~s!\Q/*\E.*\Q%include\E\s*\Q"GRAPHMON.INS"\E\s*\Q*/\E!  %include "GRAPHMON.INS"  !i;
    $line=~s!\Q/*\E.*\Q%include\E\s*\Q"GRAPHANA.INS"\E\s*\Q*/\E!  %include "GRAPHANA.INS"  !i;
    #take out PG monochromator  and analyzer
    $line=~s!\Q%include\E\s*\Q"CUMON.INS" \E!/*%include "CUMON.INS"*/!i;
    $line=~s!\Q%include\E\s*\Q"HSMON.INS" \E!/*%include "HSMON.INS"*/!i;
    $line=~s!\Q%include\E\s*\Q"CUANA.INS" \E!/*%include "CUANA.INS"*/!i;
    $line=~s!\Q%include\E\s*\Q"HSANA.INS" \E!/*%include "HSANA.INS"*/!i;
    print Fout $line;
    }
   close Fin;   close Fout;
   unless (rename "mode.out",$file)
    {print "\n error (perl rename cannot cross filesystems) \n";exit 1;}
   print ">\t";
   print "\n panda PG002 monochromator and analyzer inserted\n";
```

```
#
```

A.7 MONCU111

!/usr/bin/perl

```
# program moncu111 used to insert copper monochromator and analyzer
# into panda mcstas simulation
# usage: moncu111 (sets program file panda.ins to cu111 monochromator
# and analyzer)

   $file="PANDA.INS";
   print "<".$file;
   open (Fin, $file);
   open (Fout, ">mode.out");
   while($line=<Fin>)
    {
    #change mono parameters
    $line=~s!\QQ_mono=\E.*;!Q_mono= 3.0105;!;
    $line=~s!\Qr0_mono=\E.*;!r0_mono= 0.4;!;
    #put in copper monochromator and analyzer
    $line=~s!\Q/*\E.*\Q%include\E\s*\Q"CUMON.INS"\E\s*\Q*/\E!  %include "CUMON.INS"  !i;
    $line=~s!\Q/*\E.*\Q%include\E\s*\Q"CUANA.INS"\E\s*\Q*/\E!  %include "CUANA.INS"  !i;
    #take out PG monochromator and analyzer
    $line=~s!\Q%include\E\s*\Q"GRAPHMON.INS" \E!/*%include "GRAPHMON.INS"*/!i;
    $line=~s!\Q%include\E\s*\Q"HSMON.INS" \E!/*%include "HSMON.INS"*/!i;
    $line=~s!\Q%include\E\s*\Q"GRAPHANA.INS" \E!/*%include "GRAPHANA.INS"*/!i;
    $line=~s!\Q%include\E\s*\Q"HSANA.INS" \E!/*%include "HSANA.INS"*/!i;
    print Fout $line;
    }
   close Fin;   close Fout;
   unless (rename "mode.out",$file)
   {print "\n error (perl rename cannot cross filesystems) \n";exit 1;}
   print ">\t";
   print "\n panda cu111 monochromator and analyzer inserted\n";

#
```

A.8 MONHS

!/usr/bin/perl

```
# program monhs used to insert heusler monochromator and analyzer
# into panda mcstas simulation
# usage: monhs  (sets program file panda.ins to heusler  monochromator
# and analyzer)

   $file="PANDA.INS";
   print "<".$file;
   open (Fin, $file);
   open (Fout, ">mode.out");
   while($line=<Fin>)
    {
    #change mono parameters
    $line=~s!\QQ_mono=\E.*;!Q_mono= ????;!;
    $line=~s!\Qr0_mono=\E.*;!r0_mono= 0.3;!;
    #put in heusler monochromator and analyzer
    $line=~s!\Q/*\E.*\Q%include\E\s*\Q"HSMON.INS"\E\s*\Q*/\E!  %include "HSMON.INS"  !i;
    $line=~s!\Q/*\E.*\Q%include\E\s*\Q"HSANA.INS"\E\s*\Q*/\E!  %include "HSANA.INS"  !i;
    #take out PG,hs monochromator and analyzer
    $line=~s!\Q%include\E\s*\Q"CUMON.INS" \E!/*%include "CUMON.INS"*/!i;
    $line=~s!\Q%include\E\s*\Q"GRAPHMON.INS" \E!/*%include "GRAPHMON.INS"*/!i;
    $line=~s!\Q%include\E\s*\Q"CUANA.INS" \E!/*%include "CUANA.INS"*/!i;
    $line=~s!\Q%include\E\s*\Q"GRAPHANA.INS" \E!/*%include "GRAPHANA.INS"*/!i;
    print Fout $line;
    }
   close Fin;   close Fout;
   unless (rename "mode.out",$file)
   {print "\n error (perl rename cannot cross filesystems) \n";exit 1;}
   print ">\t";
   print "\n panda heusler monochromator and analyzerinserted\n";

#
```

A.9 SAMPLE

!/usr/bin/perl

```
$spl=$ARGV[0];
unless ($spl=~/V/i or $spl=~/R/i)
{print " program SAMPLE used to change SAMPLE of panda mcstas simulation\n";
 print " usage: SAMPLE R   -   sets program file panda.ins to RESSAMPLE  \n";
 print " V ... Vanadium sample \n";
 exit 0;}

   $file="PANDA.INS";
   print "<".$file;
   open (Fin, $file);
   open (Fout, ">spl.out");
   while($line=<Fin>)
  {
  #switch off all samples
  $line=~s!(\Q%include\E\s*\Q"RESSPL.INS"\E)\s!/*%include "RESSPL.INS"*/!i;
  $line=~s!(\Q%include\E\s*\Q"VANSPL.INS"\E)\s!/*%include "VANSPL.INS"*/!i;
  #switch on specific SAMPLE
  if ($spl=~/R/i)
  {$line=~
  s!\s*\Q/*\E.*\Q%include\E\s*\Q"RESSPL.INS"\E\s*\Q*/\E!  %include "RESSPL.INS" !i;}
  if ($spl=~/V/i)
  {$line=~
  s!\s*\Q/*\E.*\Q%include\E\s*\Q"VANSPL.INS"\E\s*\Q*/\E!  %include "VANSPL.INS" !i;}
  print Fout $line;
  }
   close Fin;
   close Fout;
   unless (rename "spl.out",$file)
     {die "\n error switching spl \n (perl rename cannot cross filesystems) \n";}
   print ">\t";

   $file="DETECTOR.INS";
   print "<".$file;
   open (Fin, $file);
   open (Fout, ">spl.out");
   while($line=<Fin>)
  {
  #switch on specific resolution monitor if necessary
  if ($spl=~/R/i)
  {$line=~s!\Q/*ressplonly\E\s+!/*ressplonly*/!g;
   $line=~s!\s+\Qonlyresspl*/\E!/*onlyresspl*/!g;}
  # switch of resolution monitor if necessary
  if ($spl=~/V/i)
    {$line=~s!\Q/*ressplonly*/\E!/*ressplonly  !g;
    $line=~s!\Q/*onlyresspl*/\E!  onlyresspl*/!g;}
  print Fout $line;
  }
   close Fin;
   close Fout;
   unless (rename "spl.out",$file)
     {die "\n error switching spl \n (perl rename cannot cross filesystems) \n";}
   print ">\t";

 print "\n panda spl ".$spl." switched on\n";

#
```

A.10 DSA

!/usr/bin/perl

```
unless ($#ARGV > 0)
{print "program dsa used to set distance sample- analyzer in panda mcstas simulation\n";
 print" usage: dsa 1.2  (sets program file panda.ins distance-sample-analyzer to 1.2m) \n";
 exit 0;}

$dd=$ARGV[0];
if ($dd > 0.6 & & $dd < 2)
 { $file="panda.ins";
   print "<".$file;
   open (Fin, $file);
   open (Fout, ">dsa.out");
   while($line=<Fin>)
    {
    #change distance sample analyzer
    $line=~s!\Qdist_sample_analyzer=\E.*;!dist_sample_analyzer=$dd;!;
    print Fout $line;
    }
   close Fin;   close Fout;
   unless (rename "dsa.out",$file)
   {print "\n error (perl rename cannot cross filesystems) \n";exit 1;}
   print ">\t";
   print "\n distance sample-analyzer set to ".$dd."m \n";
}
else
 {print "distance".$dd." out of 0.6-2m range \n";exit 0;}
fi


#
```

A.11 MCPLOTMR.PL

! /usr/bin/perl -w

```
# modified module mcplot to automatically produce postscirpt file
# use as: mcplotmr
use FileHandle;
use PDL;
use PDL::Graphics::PGPLOT;
use PGPLOT;

use lib "/usr/local/lib/mcstas";
use lib $ENV{"MCSTAS"};

require "mcfrontlib.pl";
require "MCPLTLIBMR.PL";


my ($file) = @ARGV;
$file = "mcstas.sim" unless $file;
$file = "$file/mcstas.sim" if -d $file;
my ($instr_inf, $sim_inf, $datalist) = read_sim_file($file);
die "No data in simulation file '$file'"
    unless @$datalist;

#print "Click on a plot for full-window view.\n" if @$datalist > 1;
#print "Type 'P' (in graphics window) for hardcopy, 'Q' to quit.\n";

#for(;;) {
    my ($cc,$cx,$cy,$idx);
    # Do overview plot, letting user select a plot for full-screen zoom.
    ($cc,$idx) = overview_plot("/xserv", $datalist, 0);
#    last if $cc =~ /[xq]/i; # Quit?
#    if($cc =~ /[pc]/i) { # Hardcopy?
# my $dev = ($cc =~ /c/i) ? "cps" : "ps";
      #overview_plot("mcstas.ps/$dev", $datalist, 0);

# change cps to ps if you do not want color postscript

overview_plot("mcstas.ps/cps", $datalist, 0);
print "Wrote postscript file 'mcstas.ps'\n";
# next;
#    }

    # now do a full-screen version of the plot selected by the user.
#    ($cc, $cx, $cy) = single_plot("/xserv", $datalist->[$idx], 1);
#    last if $cc =~ /[xq]/i; # Quit?
#    if($cc =~ /[pc]/i) { # Hardcopy?
# my $dev = ($cc =~ /c/i) ? "cps" : "ps";
# my $filename = "$datalist->[$idx]{'Component'}.ps";
# single_plot("$filename/$dev", $datalist->[$idx], 0);
# print "Wrote postscript file '$filename'\n";
#                       }
#}
#
```

A.12 MCRESPLOTMR.PL

! /usr/bin/perl

```
## modifier mcresplot to create postscript file


use PDL;
use PDL::Math;
use PDL::Slatec;
use PDL::IO::FastRaw;
use PGPLOT;
use PDL::Graphics::TriD;
use PDL::Graphics::PGPLOT;

$PI = 3.14159265358979323846;

sub read_mcstas_res {
    my ($filename) = @_;
    my ($data,$kix,$kiy,$kiz,$kfx,$kfy,$kfz,$x,$y,$z,$pi,$pf);
    my ($size,$ki,$kf,$q,$qx,$qy,$qz,$p,$Ei,$Ef,$w);
    my ($r,$qx_mc,$qy_mc,$qz_mc,$w_mc, $npts,$cntr,$gaus);
    my ($ave_q,$unit_q,$unit_n,$unit_z,$tmat,$q_t);
    my ($A,$ave_A,$mid_A,$C,$umat,$C_t,$res_mat);
    my ($pos);

    # Read data from file (either raw or ascii).
    if($filename =~ /\.raw$/) {
$data = readfraw($filename);
($kix,$kiy,$kiz,$kfx,$kfy,$kfz,$x,$y,$z,$pi,$pf) = dog $data;
    } else {
($kix,$kiy,$kiz,$kfx,$kfy,$kfz,$x,$y,$z,$pi,$pf) = rcols($filename);
$data = cat ($kix,$kiy,$kiz,$kfx,$kfy,$kfz,$x,$y,$z,$pi,$pf);
    }
    # Compute some basic entities
    ($size) = $kix->dims;
    $ki = cat($kix, $kiy, $kiz);
    $kf = cat($kfx, $kfy, $kfz);
    $q = $ki - $kf;
    $Ei = 2.072*($kix*$kix+$kiy*$kiy+$kiz*$kiz);
    $Ef = 2.072*($kfx*$kfx+$kfy*$kfy+$kfz*$kfz);
    $w = $Ei-$Ef;
    $p = $pi*$pf;
    # Compute coordinate change: X along average Q vector projected
    # into plane, Y perpendicular to X in plane, Z upwards.
    $ave_q = sumover($q*$p->dummy(1,3)) / (sum($p));
    $unit_q = $ave_q->copy;
    $unit_q->set(1,0); # Force into scattering plane.
    $unit_q /= sqrt(inner($unit_q,$unit_q));
    $unit_n = pdl($unit_q->at(2), 0, -$unit_q->at(0));
    $unit_z = pdl(0,1,0);
    # Build ortogonal transformation matrix, and change coordinates of Q.
    $tmat = cat ($unit_q, $unit_n, $unit_z);
    $q_t = clump(xchg(PDL::Primitive::matmult($tmat,$q->dummy(2)),1,2),2);
    ($qx,$qy,$qz) = dog $q_t;

    # Now compute resolution matrix.
    $A = append($q->transpose, $w->dummy(0));
    $ave_A = sumover($A->transpose*$p->dummy(1,4)) / sum($p);
    $mid_A = $A - $ave_A->dummy(1);
    # Get the covariance matrix in original coordinates.
    $C = PDL::Primitive::matmult
($mid_A->transpose, $mid_A*$p->dummy(0,4)) / sum($p);
    # Change coordinates, and compute the resolution matrix.
    $umat = transpose(append(transpose(append($tmat,pdl [0])),
     pdl [[0],[0],[0],[1]]));
    $C_t = inner2t($umat->transpose,$C,$umat);
    $res_mat = $C_t->matinv;
    print "The covariance matrix is\n";
    print $C_t;
    print "and the resolution matrix is\n";
    print $res_mat;

    # Plot histograms for the four 1-d projections.
    dev "/cps", 4, 3;
    pgsch(2.1);
    pgsci(1);
    q_hist($qx, $p, "Q\\dx\\u [\\A\\u-1\\d]");
    q_hist($qy, $p, "Q\\dy\\u [\\A\\u-1\\d]");
    pgpage;
    pgmtxt("t",-1,0.0,0.0,"Resolution matrix:");
    $pos = matout(-3.0, $res_mat);
    pgmtxt("t",$pos-1,0.0,0.0,"Bragg Widths");
    pgmtxt("t",$pos-3-0*1.2,0.2,0.0,"Q\\dx\\u = " .
   2.3548/sqrt($res_mat->at(0,0)));
    pgmtxt("t",$pos-3-1*1.2,0.2,0.0,"Q\\dy\\u = " .
   2.3548/sqrt($res_mat->at(1,1)));
    pgmtxt("t",$pos-3-2*1.2,0.2,0.0,"Q\\dz\\u = " .
   2.3548/sqrt($res_mat->at(2,2)));
    pgmtxt("t",$pos-3-3*1.2,0.2,0.0,"\\gw = " .
   2.3548/sqrt($res_mat->at(3,3)));
    pgmtxt("t",$pos-9,0.0,0.0,"Covariance matrix:");
    $pos = matout($pos-11.0, $C_t);
    pgpage;
    q_hist($qz, $p, "Q\\dz\\u [\\A\\u-1\\d]");
    q_hist($w,  $p, "\\gw [meV]");
    pgpage;pgpage;
    mcs_projs(1,$res_mat);

#    # Make a 3d visualization of the resolution elipsoid. Use MC
#    # choice to eliminate the weights.
#    $r = random $size;
#    $qx_mc = $qx->where($p > $r*max($p));
#    $qy_mc = $qy->where($p > $r*max($p));
#    $qz_mc = $qz->where($p > $r*max($p));
#    $w_mc = $w->where($p > $r*max($p));
#    $npts = $w_mc->nelem;
#    print "\nPress 'Q' in 3D window to continue ...\n";
#    points3d [$qx_mc,$qy_mc,$w_mc],
#      [0.8*ones($npts),0.8*ones($npts),zeroes($npts)];
#
#    # Now overlay with points generated from the corresponding gaussian.
#    $cntr = PDL::Primitive::matmult($umat,$ave_A->dummy(0));
#    $gaus = PDL::Primitive::matmult(chol($C_t),grandom($npts,4)) + $cntr;
#    print "Adding gausian distribution\nPress 'Q' in 3D window to end ...\n";
#    hold3d();
#    points3d [$gaus->slice(",(0)"),$gaus->slice(",(1)"),$gaus->slice(",(3)")],
#      [0.9*ones($npts),zeroes($npts),zeroes($npts)];
#    release3d();
}

sub chol {
    my ($A) = @_;
    my ($i,$j,$k,$L,$n,$n2,$li,$lj,$v);

    ($n,$n2) = $A->dims;
    die "Must be square matrix" unless $n==$n2;
    $L = zeroes $n,$n;
    $li = $lj = pdl []; # Handle special case for i=0
    for($i=0; $i<$n; $i++) {
$li = $L->mslice([0,$i-1],[$i]) if $i;
$v = $A->at($i,$i) - sum($li*$li);
die "Not positive definite" unless $v >= 0;
$L->set($i,$i, sqrt($v));
for($j=$i+1; $j<$n; $j++) {
    $lj = $L->mslice([0,$i-1],[$j]) if $i;
    $L->set($i,$j, ($A->at($i,$j) - sum($li*$lj))/$L->at($i,$i));
}
    }
    return $L;
}

sub q_hist {
    my ($v,$p,$t) = @_;
    my ($m,$h) = weighted_hist($v,$p,50);
    bin(($m,$h/max($h)),{COLOUR => YELLOW});
    pglab($t,"R(Q,\\gw) [A.U]", "");
}

sub weighted_hist {
    my ($x,$p,$n) = @_;
    my ($xmin,$xmax,$h,$bin,$xmid);

    ($xmin,$xmax) = minmax($x);
    $h = zeroes $n;
    $bin = long(($x - $xmin)/($xmax - $xmin)*$n);
    $xmid = sequence($n)/$n*($xmax - $xmin) + $xmin + ($xmax - $xmin)/(2*$n);
    for(0..($n-1)) {
$h->set($_, sum($p->where($bin==$_)));
    }
    $h->set($n-1, $h->at($n-1) + sum($p->where($bin==$n)));
    return ($xmid, $h);
}

sub matout {
    my ($pos,$x) = @_;
    my @lines = split("\n","$x");
#    shift(@lines);shift(@lines);pop(@lines);
    for(@lines) {
if(m’\[([^]]*)\]’) {
    pgmtxt("t",$pos,0.0,0.0,$1);
    $pos-= 1.2;
}
    }
    return $pos;
}

# The rest of this file is converted from rescal5 matlab code.
sub rot_elip {
    my ($a,$b,$phi) = @_;
    my($n,$x,$y,$s,$c,$th);

    $n = 100;
    $th = sequence($n+1)/$n*2*$PI;
    $x = $a*cos($th);
    $y = $b*sin($th);
    $c = cos($phi);
    $s = sin($phi);
    $th = $x*$c - $y*$s;
    $y = $x*$s + $y*$c;
    $x = $th;
    return ($x,$y);
}

sub bad_rc_int {
    my ($i,$r0,$m) = @_;
    my ($n1,$n2,$r,$sel,$b,$mp,$new);

    ($n1,$n2) = $m->dims;
    die "Must have square input matrix" unless $n1==$n2;
    $r = sqrt(2*$PI/$m->at($i,$i))*$r0;
    $sel = pdl [0..$i-1,$i+1..$n1-1];
    $b = $m->slice(",($i)") + $m->slice("($i),");
    $b = $b->dice($sel);
    $mp = $m->dice($sel,$sel);
    $new = $mp - 1/(4*$m->at($i,$i))*
      PDL::Primitive::matmult($b->dummy(0),$b->dummy(1));
    return ($r, $new);
}

sub rc_int {
    my ($i,$r0,$m) = @_;
    my ($n1,$n2,$r,$sel,$b,$mp,$new);

    ($n1,$n2) = $m->dims;
    die "Must have square input matrix" unless $n1==$n2;
    $r = sqrt(2*$PI/$m->at($i,$i))*$r0;
    $sel = pdl [0..$i-1,$i+1..$n1-1];
    $b = $m->slice(",($i)") + $m->slice("($i),");
    $b = $b->dice($sel);

    $mp = zeroes $n1-1,$n2-1;
    if($i > 0) {
$mp = $mp->ins($m->mslice([0,$i-1],[0,$i-1]),0,0);
    }
    if($i < $n1 - 1) {
$mp = $mp->ins($m->mslice([$i+1,$n1-1],[$i+1,$n2-1]),$i,$i);
    }
    if($i > 0 && $i < $n1 - 1) {
```

```
$mp = $mp->ins($m->mslice([0,$i-1],[$i+1,$n2-1]),0,$i);
$mp = $mp->ins($m->mslice([$i+1,$n1-1],[0,$i-1]),$i,0);
    }
    $new = $mp - 1/(4*$m->at($i,$i))*
      PDL::Primitive::matmult($b->dummy(0),$b->dummy(1));
    return ($r, $new);
}

sub mcs_projs {
    my ($R0,$NP) = @_;

    mcs_proj($R0,$NP,2,"Q\\dx\\u [\\A\\u-1\\d]",
     "Q\\dy\\u [\\A\\u-1\\d]",pdl([0,1,3]),pdl([0,1]));
    mcs_proj($R0,$NP,1,"Q\\dx\\u [\\A\\u-1\\d]",
     "\\gw [meV]",pdl([0,1,3]),pdl([0,3]));
    mcs_proj($R0,$NP,0,"Q\\dy\\u [\\A\\u-1\\d]",
     "\\gw [meV]",pdl([0,1,3]),pdl([1,3]));

    mcs_proj($R0,$NP,0,"Q\\dz\\u [\\A\\u-1\\d]",
     "\\gw [meV]",pdl([0,2,3]),pdl([2,3]));
}

sub mcs_proj {
    my ($R0,$A,$index,$xlabel,$ylabel,$sel1,$sel2) = @_;
    my($B,$R0P,$MP,$x,$y);

    $B = $A->dice($sel1,$sel1);
    ($R0P,$MP) = rc_int($index,$R0,$B);
    ($x,$y) = proj_elip($MP);
    poly($x,$y, {COLOUR => RED});
    hold;
    line($x,$y,{COLOUR => BLACK});
    ($x,$y) = proj_elip($A->dice($sel2,$sel2));
    poly($x,$y, {COLOUR => GREEN});
    line($x,$y,{COLOUR => BLACK});
    pglab($xlabel,$ylabel,"");
    rel;
}

sub proj_elip {
    my ($MP) = @_;
    my ($const,$theta,$S,$MP2,$hwhm_xp,$hwhm_yp,$x,$y);

    $const = 1.17741;
    $theta = 0.5*atan(2*$MP->at(0,1)/($MP->at(0,0)-$MP->at(1,1)));
    $S = cat(cat(cos($theta),  sin($theta)),
     cat(-sin($theta), cos($theta)));
    $MP2 = inner2t($S->transpose,$MP,$S);
    $hwhm_xp=$const/sqrt($MP2->at(0,0));
    $hwhm_yp=$const/sqrt($MP2->at(1,1));
    ($x,$y) = rot_elip($hwhm_xp,$hwhm_yp,$theta);
    return ($x,$y);
}

read_mcstas_res($ARGV[0]);

1;
```

```
#
```

A.13 MCDISPMR.PL

! /usr/bin/perl -w

```
# this is mcdisplay martin rotter version
# it has been enhanced to plot out the
# instrument onto mcstas.ps on exiting the program
# martin rotter             13.7.99


# In emacs, please make this -*- perl -*- mode. Thanks.


use PGPLOT;

$magnification = 1;
$zooming = 0;

my (%transformations, @components);


sub read_instrument {
    my ($in) = @_;
    my ($st, $comp);

    $st = 0;
    @components = ();
    while(<$in>) {
if($st == 0 && /^INSTRUMENT:/) {
    # Start of instrument description.
    $st = 1;
} elsif($st == 1 && /^COMPONENT:\s*"([a-zA-Z0-9_]+)"\s*/) {
    $comp = $1;
    @components = (@components, $comp);
} elsif($st == 1 && /^POS:(.*)$/) {
    my @T;
    @T = split ",", $1;
    $transformations{$comp} = \@T;
} elsif($st == 1 && /^MCDISPLAY: start$/) {
    $st = 2; # Start of component graphics representation
} elsif($st == 2 && /^MCDISPLAY: component ([a-zA-Z0-9_]+)/) {
    $comp = $1;
    $compdraw{$comp} = {};
    $compdraw{$comp}{'elems'} = [];
} elsif($st == 2 && /^MCDISPLAY: magnify\('([xyz]*)'\)$/) {
    my $mag = $1;
    $compdraw{$comp}{'magX'} = 1 if $mag =~ /x/i;
    $compdraw{$comp}{'magY'} = 1 if $mag =~ /y/i;
    $compdraw{$comp}{'magZ'} = 1 if $mag =~ /z/i;
} elsif($st == 2 && /^MCDISPLAY: multiline\(([0-9]+),([^()\n]+)\)$/) {
    my $count = $1;
    my @coords = split ',', $2;
    push @{$compdraw{$comp}{'elems'}},
{type => 'multiline',
 count => $count,
 coords => \@coords};
} elsif($st == 2 &&
/^MCDISPLAY: circle\('([xyzXYZ]{2})',([-+0-9.eE]+),([-+0-9.eE]+),([-+0-9.eE]+),([-+0-9.eE]+)\)$/) {
    my ($plane,$x,$y,$z,$r) = ($1,$2,$3,$4,$5);
    # Make a circle using a 25-order multiline.
    my @coords = ();
    my $i;
    for($i = 0; $i <= 24; $i++) {
my $a = $r*cos(2*3.1415927/24*$i);
my $b = $r*sin(2*3.1415927/24*$i);
my ($x1,$y1,$z1) = ($x,$y,$z);
if($plane =~ /xy|yx/i) {
    $x1 += $a;
    $y1 += $b;
} elsif($plane =~ /xz|zx/i) {
    $x1 += $a;
    $z1 += $b;
} elsif($plane =~ /yz|zy/i) {
    $y1 += $a;
    $z1 += $b;
} else {
    die "Bad plane specifier in circle: '$plane'";
}
push @coords, $x1, $y1, $z1;
    }
    push @{$compdraw{$comp}{'elems'}},
{type => 'multiline',
 count => 25,
 coords => \@coords};
} elsif($st == 2 && /^MCDISPLAY: end$/) {
    $st = 1; # End of component graphics representation
} elsif($st == 1 && /^INSTRUMENT END:/) {
    $st = 100;
    last;
} else {
    print;
}
    }
    exit if($st != 100); # Stop when EOF seen before instrument end.
    return $#components + 1;
}


sub make_instrument {
    my (@x, @y, @z, @ori, @dis, @comp);
    my ($i, $c, %instr);
    my ($xmin, $xmax, $ymin, $ymax, $zmin, $zmax);

    $i = 0;
    foreach $c (@components) {
my (@T, @U);
@T = @{$transformations{$c}};
$x[$i] = $T[0];
$xmin = $x[$i] if !$xmin || $x[$i] < $xmin;
$xmax = $x[$i] if !$xmax || $x[$i] > $xmax;
$y[$i] = $T[1];
$ymin = $y[$i] if !$ymin || $y[$i] < $ymin;
```

```
$ymax = $y[$i] if !$ymax || $y[$i] > $ymax;
$z[$i] = $T[2];
$zmin = $z[$i] if !$zmin || $z[$i] < $zmin;
$zmax = $z[$i] if !$zmax || $z[$i] > $zmax;

@U = ($T[3], $T[4], $T[5], $T[6], $T[7], $T[8], $T[9], $T[10], $T[11]);
$ori[$i] = \@U;
$comp[$i] = $c;
# Now transform coordinates for component graphics representations.
if($compdraw{$c}) {
    my $magX = $compdraw{$c}{'magX'};
    my $magY = $compdraw{$c}{'magY'};
    my $magZ = $compdraw{$c}{'magZ'};
    foreach $elem (@{$compdraw{$c}{'elems'}}) {
if($elem->{'type'} eq 'multiline') {
    my @coords = @{$elem->{'coords'}};
    my @xs = ();
    my @ys = ();
    my @zs = ();
    my ($xv,$yv,$zv);
    while(@coords) {
$xv = shift(@coords);
$yv = shift(@coords);
$zv = shift(@coords);
$xv *= $magnification if $magX;
$yv *= $magnification if $magY;
$zv *= $magnification if $magZ;
push @xs, ($xv*$T[3] + $yv*$T[6] + $zv*$T[9]  + $T[0]);
push @ys, ($xv*$T[4] + $yv*$T[7] + $zv*$T[10] + $T[1]);
push @zs, ($xv*$T[5] + $yv*$T[8] + $zv*$T[11] + $T[2]);
    }
    $elem->{'X'} = \@xs;
    $elem->{'Y'} = \@ys;
    $elem->{'Z'} = \@zs;
}
    }
}
$i++;
    }
    if($xmin == $xmax) {
$xmin--;
$xmax++;
    }
    if($ymin == $ymax) {
$ymin--;
$ymax++;
    }
    if($zmin == $zmax) {
$zmin--;
$zmax++;
    }
    $xmin -= ($xmax - $xmin) / 6;
    $xmax += ($xmax - $xmin) / 6;
    $ymin -= ($xmax - $xmin) / 6;
    $ymax += ($xmax - $xmin) / 6;
    $zmin -= ($xmax - $xmin) / 6;
    $zmax += ($xmax - $xmin) / 6;
    %instr = ('x' => \@x, 'y' => \@y, z => \@z,
      ori => \@ori, dis => \@dis, comp => \@comp,
      xmin => $xmin, xmax => $xmax,
      ymin => $ymin, ymax => $ymax,
      zmin => $zmin, zmax => $zmax,
      zoom_xmin => $xmin, zoom_xmax => $xmax,
      zoom_ymin => $ymin, zoom_ymax => $ymax,
      zoom_zmin => $zmin, zoom_zmax => $zmax);
    return %instr;
}


sub transform {
    my ($comp, $x, $y, $z, $vx, $vy, $vz, $t, $ph1, $ph2) = @_;
    if(!$comp) {
return ($x, $y, $z, $vx, $vy, $vz, $t, $ph1, $ph2);
    } else {
my ($nx, $ny, $nz, $nvx, $nvy, $nvz, $nt, $nph1, $nph2);
my @T = @{$transformations{$comp}};
$x *= $magnification if $compdraw{$comp} && $compdraw{$comp}{'magX'};
$y *= $magnification if $compdraw{$comp} && $compdraw{$comp}{'magY'};
$z *= $magnification if $compdraw{$comp} && $compdraw{$comp}{'magZ'};
$nx = $x*$T[3] + $y*$T[6] + $z*$T[9] + $T[0];
$ny = $x*$T[4] + $y*$T[7] + $z*$T[10] + $T[1];
$nz = $x*$T[5] + $y*$T[8] + $z*$T[11] + $T[2];
$nvx = $vx*$T[3] + $vy*$T[6] + $vz*$T[9];
$nvy = $vx*$T[4] + $vy*$T[7] + $vz*$T[10];
$nvz = $vx*$T[5] + $vy*$T[8] + $vz*$T[11];
return ($nx, $ny, $nz, $nvx, $nvy, $nvz, $t, $ph1, $ph2);
    }
}


sub get_inspect_pos {
    my ($inspect, @comps) = @_;
    return 0 unless $inspect;
    my $i;
    for($i = 0; $i < @comps; $i++) {
return $i if $comps[$i] eq $inspect;
    }
    die "Error: Inspected component $inspect not part of instrument?";
}


sub read_neutron {
    my ($in) = @_;
    my (@x, @y, @z, @vx, @vy, @vz, @t, @ph1, @ph2, @ncomp);
    my ($st, $i);
    my $comp;

    $st = 0;
    $i = 0;
    my $dropit = 1; # Flag to drop uninteresting neutron states.
    while(<$in>) {
if($st == 0 && /^ENTER:/) {
    # Neutron enters instrument.
    $st = 1;
} elsif($st == 0 && /^STATE:/) {
    # State after leaving - should probably be removed in McStas.
    next;
} elsif($st == 1 && /^COMP:\s*"([a-zA-Z0-9_]+)"\s*$/) {
    # Neutron enters component local coordinate system.
    $comp = $1;
    $dropit = 1; # Drop the first state (entry point).
} elsif($st == 1 && /^STATE:(.*)$/) {
    # Neutron state.
    $dropit = 0, next if $dropit; # Skip entry point
    ($x[$i], $y[$i], $z[$i],
     $vx[$i], $vy[$i], $vz[$i],
     $t[$i], $ph1[$i], $ph2[$i]) = split ",", $1;
    ($x[$i], $y[$i], $z[$i],
     $vx[$i], $vy[$i], $vz[$i],
     $t[$i], $ph1[$i], $ph2[$i]) =
 transform($comp, $x[$i], $y[$i], $z[$i],
   $vx[$i], $vy[$i], $vz[$i],
   $t[$i], $ph1[$i], $ph2[$i]);
    $ncomp[$i] = $comp;
    $i++;
} elsif($st == 1 && /^ABSORB:/) {
    # Neutron was absorbed.
    next; # No special action needed.
} elsif($st == 1 && /^LEAVE:/) {
    # Neutron leaves instrument.
    $st = 2;
    last;
} else {
    # Pass on any output not meant for us.
    print;
}
    }
    exit unless $st == 2; # Stop when EOF seen before neutron data end.

    my %neutron = (’x’ => \@x, ’y’ => \@y, z => \@z,
   vx => \@vx, vy => \@vy, vz => \@vz,
   t => \@t, ph1 => \@ph1, ph2 => \@ph2, comp => \@ncomp);
    return %neutron
}


sub plot_components {
    my ($rx, $ry, $rori, $rdis, $axis1, $axis2) = @_;
    my (@x, @y, @ori);
    my ($i, $col);

    @x = @$rx;
    @y = @$ry;
    @ori = @$rori;

    pgsci(2);
    pgline($#x + 1, \@x, \@y);
    pgpt($#x + 1, \@x, \@y, 20);
    $col = 4;
    for($i = 0; $i <= $#components; $i++) {
my $comp = $components[$i];
```

```
pgsci($col++);
$col = 4 if $col > 15;
if($compdraw{$comp}) {
    foreach $elem (@{$compdraw{$comp}{'elems'}}) {
if($elem->{'type'} eq 'multiline') {
    pgline($elem->{'count'}, $elem->{$axis1}, $elem->{$axis2});
}
    }
} else {
    pgsch(1.4);
    pgpt(1, $x[$i], $y[$i], 26);
}
    }
}


sub plot_neutron {
    my ($rx, $ry, $rvx, $rvy) = @_;
    my (@x, @y);
    my ($i, $col);

    @x = @$rx;
    @y = @$ry;
    pgsci(3);
    pgline($#x + 1, \@x, \@y);
    # Show component entry/exit points in same colour as respective component.
    $i = 0;
    $col = 4;
    while($i <= $#x) {
pgsci($col++);
$col = 4 if $col > 15;
# Exit point.
pgpt(1, $x[$i], $y[$i], 17);
$i++;
    }
}


sub show_comp_names {
    my ($rinstr) = @_;
    my %instr = %$rinstr;
    my @comps = @{$instr{'comp'}};
    my $count = @comps;
    $count = 8 if $count < 8;
    my $col = 4;
    pgsch(25/$count);
    my $i;
    for($i = 0; $i < @comps; $i++) {
pgsci($col++);
$col = 4 if $col > 15;
pgmtxt('RV', 0.2, 1 - ($i+0.5)/$count, 0.0, $comps[$i]);
    }
}


sub reset_zoom {
    my ($rinstr, $vps) = @_;
    $rinstr->{’zoom_xmin’} = $rinstr->{’xmin’};
    $rinstr->{’zoom_xmax’} = $rinstr->{’xmax’};
    $rinstr->{’zoom_ymin’} = $rinstr->{’ymin’};
    $rinstr->{’zoom_ymax’} = $rinstr->{’ymax’};
    $rinstr->{’zoom_zmin’} = $rinstr->{’zmin’};
    $rinstr->{’zoom_zmax’} = $rinstr->{’zmax’};
    $zooming = 0;
}


sub do_zoom {
    my ($rinstr, $vps, $cx, $cy, $cx1, $cy1) = @_;
    my ($tmp, $a1, $a2);
    $tmp = $cx, $cx = $cx1, $cx1 = $tmp if $cx > $cx1;
    $tmp = $cy, $cy = $cy1, $cy1 = $tmp if $cy > $cy1;

    if($cx == $cx1 || $cy == $cy1) {
print STDERR "Warning: bad zoom area.\n";
return;
    }
    if($multi_view) {
# Convert from screen coordinates to world coordinates.
# First find which of the three views was choosen.
if($cx < 0 && $cy < 1) {
    $cx = $cx + 1;
    $cx1 = $cx1 + 1;
    ($a1,$a2) = ("z", "y");
} elsif($cx < 0 && $cy >= 1) {
    $cx = $cx + 1;
    $cx1 = $cx1 + 1;
    $cy = $cy - 1;
    $cy1 = $cy1 - 1;
    ($a1,$a2) = ("z", "x");
} elsif($cx >= 0 && $cy >= 1) {
    $cy = $cy - 1;
    $cy1 = $cy1 - 1;
    ($a1,$a2) = ("x", "y");
} else {
    print STDERR "Warning: bad zoom area.\n";
    return;
}
my $idx = "$a1-$a2";
my $vpx0 = $vps->{$idx}{’VP’}[0];
my $vpdx = $vps->{$idx}{’VP’}[1] - $vpx0;
my $wx0 = $vps->{$idx}{’W’}[0];
my $wdx = $vps->{$idx}{’W’}[1] - $wx0;
my $vpy0 = $vps->{$idx}{’VP’}[2];
my $vpdy = $vps->{$idx}{’VP’}[3] - $vpy0;
my $wy0 = $vps->{$idx}{’W’}[2];
my $wdy = $vps->{$idx}{’W’}[3] - $wy0;
$cx = ($cx-$vpx0)/$vpdx*$wdx+$wx0;
$cx1 = ($cx1-$vpx0)/$vpdx*$wdx+$wx0;
$cy = ($cy-$vpy0)/$vpdy*$wdy+$wy0;
$cy1 = ($cy1-$vpy0)/$vpdy*$wdy+$wy0;
    } else {
($a1, $a2) = ("z","x");
    }
    $rinstr->{"zoom_${a1}min"} = $cx;
    $rinstr->{"zoom_${a1}max"} = $cx1;
    $rinstr->{"zoom_${a2}min"} = $cy;
    $rinstr->{"zoom_${a2}max"} = $cy1;
    $zooming = 1;
}


sub plot_instrument {
    my ($rinstr, $rneutron) = @_;
    my %instr = %$rinstr;
    my %neutron = %$rneutron;
    my ($xmin, $xmax, $ymin, $ymax, $zmin, $zmax) =
($instr{'zoom_xmin'}, $instr{'zoom_xmax'}, $instr{'zoom_ymin'},
 $instr{'zoom_ymax'}, $instr{'zoom_zmin'}, $instr{'zoom_zmax'});
    my %vps; # Viewport/window setup.
    my ($vpx1,$vpx2,$vpy1,$vpy2,$wx1,$wx2,$wy1,$wy2);

    pgbbuf;

    # First show instrument from "above" (view in direction of y axis).
    pgsci(1);
    pgsch(1.4);
    pgenv($zmin, $zmax, $xmin, $xmax, ($zooming ? 0 : 1), 0);
    pglab("Z Axis [m]", "X Axis [m]", "Z-X view");
    show_comp_names($rinstr);
    pgsch(1.4);
    plot_components($instr{'z'}, $instr{'x'}, $instr{'ori'}, $instr{'dis'},
    'Z', 'X');
    plot_neutron($neutron{'z'}, $neutron{'x'}, $neutron{'vz'}, $neutron{'vx'});

    if($multi_view) {
# Remember viewport setup for Z-X view.
pgqvp(0, $vpx1, $vpx2, $vpy1, $vpy2);
pgqwin($wx1, $wx2, $wy1, $wy2);
$vps{'z-x'} = {VP => [$vpx1,$vpx2,$vpy1,$vpy2],
       W => [$wx1,$wx2,$wy1,$wy2]};

# Now show instrument viewed in direction of z axis.
pgsci(1);
pgsch(1.4);
pgenv($xmin, $xmax, $ymin, $ymax, ($zooming ? 0 : 1), 0);
pglab("X Axis [m]", "Y Axis [m]", "X-Y view");
plot_components($instr{'x'}, $instr{'y'}, $instr{'ori'}, $instr{'dis'},
'X', 'Y');
plot_neutron($neutron{'x'}, $neutron{'y'}, $neutron{'vx'}, $neutron{'vy'});
# Remember viewport setup for Z-X view.
pgqvp(0, $vpx1, $vpx2, $vpy1, $vpy2);
pgqwin($wx1, $wx2, $wy1, $wy2);
$vps{'x-y'} = {VP => [$vpx1,$vpx2,$vpy1,$vpy2],
       W => [$wx1,$wx2,$wy1,$wy2]};

# Now show instrument viewed in direction of x axis.
pgsci(1);
pgsch(1.4);
pgenv($zmin, $zmax, $ymin, $ymax, ($zooming ? 0 : 1), 0);
pglab("Z Axis [m]", "Y Axis [m]", "Z-Y view");
```

```
plot_components($instr{'z'}, $instr{'y'}, $instr{'ori'}, $instr{'dis'},
'Z', 'Y');
plot_neutron($neutron{'z'}, $neutron{'y'}, $neutron{'vz'}, $neutron{'vy'});
# Remember viewport setup for Z-Y view.
pgqvp(0, $vpx1, $vpx2, $vpy1, $vpy2);
pgqwin($wx1, $wx2, $wy1, $wy2);
$vps{'z-y'} = {VP => [$vpx1,$vpx2,$vpy1,$vpy2],
       W => [$wx1,$wx2,$wy1,$wy2]};

# Set up viewport & window for mouse zoom.
pgpage;
pgsvp(0,1,0,1);
pgswin(0,1,0,1);
    }
    pgebuf;

    # Now wait for a keypress in the graphics window.
    my ($cx, $cy, $cc);
    $cx = $cy = 0;
    pgband(0, 0, 0, 0, $cx, $cy, $cc);
    if($cc =~ /[qQ]/) {
return 2; # Finished.
    } elsif($cc =~ /[zZdD]/) { # Zoom.
my ($cx1, $cy1, $cc1) = (0, 0, 0);
pgband(2,0,$cx,$cy,$cx1,$cy1,$cc1);
do_zoom($rinstr, \%vps, $cx, $cy, $cx1, $cy1);
return 1;
    } elsif($cc =~ /[xX]/) { # Reset zoom.
reset_zoom($rinstr, \%vps);
return 1;
    }
    return 0; # Default: do not repeat this neutron.
}



# Test the code.

# Attempt to locate pgplot directory if unset.
$ENV{'PGPLOT_DIR'} = "/usr/local/pgplot" unless $ENV{'PGPLOT_DIR'};

# Check command line arguments.

undef $inspect;
for(;;) {
    if(($ARGV[0] eq "-m") || ($ARGV[0] eq "--multi")) {
$multi_view = 1;
shift;
    } elsif(($ARGV[0] =~ /^-z([-0-9+.eE]+)$/) ||
    ($ARGV[0] =~ /^--zoom=([-0-9+.eE]+)$/)) {
$magnification = ($1 == 0 ? 1 : $1);
shift;
    } elsif(($ARGV[0] =~ /^-i([a-zA-Z0-9_]+)$/) ||
    ($ARGV[0] =~ /^--inspect=([a-zA-Z0-9_]+)$/)) {
$inspect = $1;
shift;
    } else {
```

```
last;
    }
}

if($multi_view) {
    # We use a 2x2 display format to view the instrument from three angles.
    pgbegin(0, "/xserv", 2, 2);
} else {
    # We use a 1x1 display for detail.
    pgbegin(0, "/xserv", 1, 1);
}
pgask(0);

my ($numcomp, %neutron, %instr, $checkexit);

$checkexit = 0;
$sim_cmd = shift;
$args = join(" ", @ARGV);
$cmdline = "$sim_cmd --trace $args";
printf STDERR "Starting simulation ’$cmdline’ ...\n";
open(IN, "$cmdline |") || die "Could not run simulation\n";

$numcomp = read_instrument(IN);
$inspect_pos = get_inspect_pos($inspect, @components);
%instr = make_instrument;

while($checkexit != 2) {
    %neutron = read_neutron(IN);
    next if @{$neutron{’comp’}} <= $inspect_pos;

    $checkexit = 1;
    while($checkexit  == 1)
      {$checkexit = plot_instrument(\%instr, \%neutron);
       }
}

#close(IN);

pgend;

if($multi_view) {
    # We use a 2x2 display format to view the instrument from three angles.
    pgbegin(0, "mcstas.ps/cps", 2, 2);
} else {
    # We use a 1x1 display for detail.
    pgbegin(0, "mcstas.ps/cps", 1, 1);
}
pgask(0);

#my ($numcomp, %neutron, %instr);


#$sim_cmd = shift;
#$args = join(" ", @ARGV);
#$cmdline = "$sim_cmd --trace $args";
#printf STDERR "Starting simulation ’$cmdline’ ...\n";
#open(IN, "$cmdline |") || die "Could not run simulation\n";
```

```
#$numcomp = read_instrument(IN);
#$inspect_pos = get_inspect_pos($inspect, @components);
#%instr = make_instrument;

#while(!eof(IN)) {
    %neutron = read_neutron(IN);
    next if @{$neutron{'comp'}} <= $inspect_pos;

    while(plot_instrument(\%instr, \%neutron))
    { }
#}

close(IN);

pgend;
#
```

B McStas Program files

B.1 PANDA.INS

```
/*

*/

DEFINE INSTRUMENT PANDA(DIAPH1,NEGLiPOSEi,NEGLfPOSEf,DNEGLPOSE,Q)

/*********************************************************************
DIAPH1... (m)horizontal diaphragm witdth at beam entry of monochromator
NEGLiPOSEi ...if negative -monochromator wavelength [A],
              if positive- monochromator energy [meV]
NEGLfPOSEf ...if negative -analyzer wavelength [A],
              if positive- analyzer energy [meV]
DNEGLPOSE  ...if negative -src/monitor  wavelength interval [A],
              if positive-src/monitor  energy interval [meV]
Q          ... scattering wave vector (1/A), Q=2pi*abs(h*h/a/a+k*k/b/b+l*l/c/c)

.....  we calculate angles A1 A2 from input parameters  ki or Ei  ....
 A1       monochromator angle theta
 A2       monochromator angle 2theta

the spectrometer has 3 different modes - one without and two with
collimator, to change from mode1 to mode2 change the comments m1m2m33m2m1m
in "panda.ins" and the include files "guide*.ins" ...
m1: no collimator, horizontal focussing mode revolver 1
m2: focussing heusler mode revolver 2   --- not fully implemented
m3: collimator, normal collimated beam mode revolver 3, collimator in analyzer
m4: collimated heusler mode
m5: primary collimated heusler (with neutron guide) analyzer focussing
**********************************************************************/

DECLARE
%{
double src_flux,mean_energy,d_E,E_in,lambda_in;  /*for the source*/
double dia1_xmin, dia1_xmax, dist_lastwin_dia1;  /*for the guide*/
double Q_mono,r0_mono,R_H,R_V,A1,A2;   /*for the focussing monochromator*/
double E_min,E_max,L_min,L_max,
       ninetyminusA1,minusA2,minusA1;  /*for the monitors */
double E_f,lambda_f,ki,kf,A3,A4,targetx,targetz;  /* for the sample*/
double dist_sample_analyzer,dist_sample_analyzer_1,
       dist_sample_analyzer_10,dist_sample_dia,
       dist_sample_mon,A5,A6,Q_ana,r0_ana,R_Hana,R_Vana,
       dist_sample_analyzer_screening_outside,
       ninetyminusA5,minusA5,minusA6; /*for analyzer*/
double dist_analyzer_detector,dist_analyzer_coll4; /* for detector*/
%}
/***********************************************************************/
INITIALIZE
%{
/**********************************************************************/
/**************   PARAMETERS  *****************************************/
/**********************************************************************/

/**************** SOURCE **********************************************/
     /* source parameters */
 src_flux= 4.5e12;  /* n/s/cm^2/AA   */
if (NEGLiPOSEi < 0 )
{lambda_in = -NEGLiPOSEi;
E_in = 81.81/(lambda_in * lambda_in);}
else
{E_in=NEGLiPOSEi;
 lambda_in=sqrt(81.81/E_in);}

 mean_energy=E_in; /* (40.5 FOR REALISTIC ENERGY PROFILE AT MONOCHROMATOR!!!)
                        meV center of rectangular energy distr*/
if(DNEGLPOSE > 0)
  {   d_E = DNEGLPOSE/2;} /*  (40 FOR REALISTIC ENERGY PROFILE AT MONOCHROMATOR!!!!)*/
else                         /*   mev  width of rectangular energy distr*/
  {   d_E =-2*E_in* DNEGLPOSE/lambda_in/2;}

/*********************** GUIDE *******************************************/
        /* guide parameters */
 dia1_xmin = -DIAPH1/2;           /* (m) diaphragm1 width /2 */
 dia1_xmax = DIAPH1/2;


/******************** MONOCHROMATOR **************************************/
        /* AN MONOCHROMATOR ANPASSEN !!!!!!!!!!!!!!!*/
Q_mono= 1.8727;  /*monochromator Q [1/A]: 3.0105 for Cu111,
                                          1.8727 for graphit 002
  3.7466 for graphit 004*/
r0_mono= 0.87;   /* monochromator reflectivity: 0.4 for Cu111,
                                                0.87 for graphit 002,
                  0.5 for graphit 004*/

A1=180*asin(lambda_in*Q_mono/4/PI)/PI;
A2=2*A1;
/*m1  R_H=-2.000/(sin(A1*PI/180));  1m*/
                              /*horizontal focussing radius of monocromator ...
                                          2.000=distance sample monochromator*/
/*m2  R_H=-2.000/(sin(A1*PI/180));  2m*//*not really needed for heuslermono*/
/*m3*/R_H=-10.000;/*3m*/        /*... no horizontal focus.  in mode 3*/
/*m4  R_H=-10.000;  4m*//*not really needed for heuslermono*/
```

```
/*m5  R_H=-2.000/(sin(A1*PI/180));  5m*//*not really needed for heuslermono*/

R_V=-2*8.000*2.000*sin(A1*PI/180)/(2.000+8.000);
                             /*vertical focussing radius of monocromator ...
                  2.000=distance sample monochromator*/
                    /* negative radii because monochromator is tilted minusa1*/
ninetyminusA1=90-A1;
minusA1=-A1;
minusA2=-A2;

/********************** SAMPLE ****************************************/
/*  monitor parameters */
E_min=mean_energy-d_E;
E_max=mean_energy+d_E;
L_min=sqrt(81.81/(mean_energy+d_E));
L_max=sqrt(81.81/(mean_energy-d_E));
/* sample parameters */
if (NEGLfPOSEf < 0 )
{lambda_f = -NEGLfPOSEf;
E_f = 81.81/(lambda_f * lambda_f);}
else
{E_f=NEGLfPOSEf;
 lambda_f=sqrt(81.81/E_f);}  /*final energy*/
ki=2*PI/lambda_in;
kf=2*PI/lambda_f;
A4=180/PI*acos(-(Q*Q-kf*kf-ki*ki)/2/kf/ki);
A3=A4/2;
 dist_sample_analyzer = 0.800 ; /*distance of sample and analyzer */
 dist_sample_analyzer_1 = dist_sample_analyzer-0.001 ;
 dist_sample_analyzer_10 = dist_sample_analyzer-0.1 ;
               /*distance of sample and analyzer beam profile monitors */
 dist_sample_mon=dist_sample_analyzer-0.22; /*distance between sample and
                                  monitor in analyzer unit*/
 dist_sample_dia=dist_sample_analyzer-0.25; /*distance between sample and
                                  diaphragm in analyzer unit*/
 dist_sample_analyzer_screening_outside=dist_sample_dia-0.25;
targetz=dist_sample_analyzer*cos(PI/180*A3) ;
          /*target position (analyser) for scattered neutrons in ressample
            coordiante system*/
targetx=dist_sample_analyzer*sin(PI/180*A3);

/****************** ANALYZER ***************************************/
/****************** DETECTOR **************************************/
 /*distance of analyzer and detector */
/*m1  dist_analyzer_detector = dist_sample_analyzer ;  1m*/
/*m2  dist_analyzer_detector = dist_sample_analyzer;  2m*/
/*m3*/dist_analyzer_detector = 0.74 ;/*3m*/
/*m4  dist_analyzer_detector = 0.74 ;  4m*/
/*m5  dist_analyzer_detector = dist_sample_analyzer;  5m*/

dist_analyzer_coll4=dist_analyzer_detector-0.201;

     /* fit to  ANALYZER !!!!!!!!!!!!!!!*/
Q_ana=1.8727;  /*monochromator Q [1/A]: 3.0105 for Cu111,
                                      1.8727 for graphit 002
3.7466 for graphit 004*/
r0_ana=0.87; /* monochromator reflectivity: 0.4 for Cu111,
                                            0.87 for graphit 002,
    0.5 for graphit 004*/
A5=180*asin(lambda_f*Q_ana/4/PI)/PI;
A6=2*A5;
/*m1  R_Hana=-dist_sample_analyzer/(sin(A5*PI/180));  1m*/
/*m2  R_Hana=-dist_sample_analyzer/(sin(A5*PI/180));  2m*/
/*m3*/R_Hana=-10.000;/*3m*/
/*m4  R_Hana=-10.000;  4m*/
/*m5  R_Hana=-dist_sample_analyzer/(sin(A5*PI/180));  5m*/
 /*horizontal focussing radius of analyzer ... */
R_Vana=-2*dist_sample_analyzer*dist_analyzer_detector*
     sin(A5*PI/180)/(dist_sample_analyzer+dist_analyzer_detector);
   /*vertical focussing radius of analyzer ...*/
                    /* negative radii because analyzer is tilted minusa5*/
ninetyminusA5=90-A5;
minusA5=-A5;
minusA6=-A6;


%}
/**********************************************************************/

TRACE
/*********************************************************************/
/*********************************************************************/
/*********************************************************************/
/*********************************************************************/

/*now we put together the instrument*/

/*********************************************************************/
/*********************************************************************/
/*********************************************************************/
/*********************************************************************/

/*********************THE SOURCE ***********************************/
%include  "SRCFLUX.INS"  /*for flux estimations*/
/*%include  "SRCADAPT.INS"*/ /* adaptive source for resolution ellipsoid test.*/

 COMPONENT shutternase = Arm() AT (0,0,0.5) ABSOLUTE  /*shutternase*/
 COMPONENT a2 = Arm()  AT (0, 0, 1.680) ABSOLUTE      /*entrance shutter*/

  /*%include "GUIDES1.INS"*/    /*with vertical and horizontal supermirror*/
  /*%include "GUIDES2.INS"*/    /* only horizontal supermirror*/
  /*%include "GUIDES3.INS"*/    /* no supermirrors*/
  /*%include "GUIDES4.INS"*/       /* all horizontal supermirrors, vertical only in
                                    inner fix part of shutter*/
  %include "GUIDES5.INS"     /* all horizontal supermirrors, vertical only
                                 in inner  fix part and revolver guide*/

/*diaphragm*/
COMPONENT dia1 = Slit(
xmin=dia1_xmin,xmax=dia1_xmax,
ymin = -0.100, ymax = 0.100)
  AT (0,0,5.880) RELATIVE shutternase ROTATED (0,0,0) RELATIVE shutternase
```

```
/********************THE MONOCHROMATOR******************************/

/*detectors to measure incoming neutrons on the monochromator (see PART I)*/
%INCLUDE "MONDET.INS"

COMPONENT A2arm = Arm()
  AT (0,0,7.880) RELATIVE shutternase ROTATED (0,minusA2,0) RELATIVE shutternase


/*CHANGE ALSO variable Q_mono according to monochromator used !!!!!!!!!!!!*/
                %include "GRAPHMON.INS"           /* graphit monochromator*/
          /*%include "CUMON.INS"*/             /* copper monochromator*/
                /*%include "HSMON.INS"*/                /* heusler monochromator*/

%include "MON2SPLE.INS"  /* some filter(not implemented), convergent
neutron guid in mode 2, coll2 */


/******************THE SAMPLE**************************************/

%include "SPLFLDET.INS"  /*detectors of sample flux,energy and position
profile(see PART I)*/

END

  /*%in*clude "VANSPL.INS"*/ /*VANADIUM SAMPLE*/
  /*%in*clude "RESSPL.INS"*/   /* sample homogenous in Q,E for calculation of
                           resolution function*/


 /*%in*clude "SPL2ANA.INS"*/ /*some slits, coll3, and detectors befor analyzer*/

 /******************THE ANALYZER***********************************/
 /*CHANGE ALSO variable Q_mono according to analyzer used !!!!!!!!!!!!*/
 /*  %in*clude "GRAPHANA.INS"  */
 /* %in*clude "CUANA.INS"*/
      /*%in*clude "HSANA.INS"*/


 /******************THE DETECTOR**********************************/
 /*%in*clude "DETECTOR.INS"*/

/*END*/

/*

*/
```

B.2 SRCADAPT.INS

```
/*

**************************************************************************
 adaptive source components of  INSTRUMENT PANDA
           - to be included into the main file panda.instr
             using %INCLUDE "srcadapt.ins"
 usage: calculation of the resolution function of the spectrometer

used parameters:
 mean_energy (meV) ...center of rectangular energy distr
 d_E         (mev) ... width/2 of rectangular energy distr
 src_flux (n/s/cm^2/AA)
*****************************************************************************/

/* this is the arm where the source is sitting on */
COMPONENT a1 = Arm()  AT (0,0,0.3) ABSOLUTE  /* from reactor centre */

COMPONENT source = Source_adapt(
        xmin=-0.068,xmax=0.068,
        ymin = -0.0400, ymax = 0.0400,  /*area of strahlrohrnase*/
        dist = 1.380,                   /*distance to A2 ... entrance shutter*/
        xw = 0.105, yh = 0.098,
                                /* ENSRC     meV energy of source 1-6.3 A*/
        E0 = mean_energy,       /*  meV center of rectangular energy distr*/
        dE = d_E,               /*  mev  width/2 of rectangular energy
                                 distribution-difference to source_disc !!!! */
        flux= src_flux,  /* n/s/cm^2/AA */
        N_E=8,   /* noofbins in energy dimension */
        N_xpos=5, /* noofbins in horizontal position*/
        N_xdiv=5, /* noofbins in horizontal divergence*/
        alpha=0.5, /*learning cut-off factor 0<alpha<=1*/
        beta=0.5, /*agressiveness of adaptive algorthm 0<beta<=1*/
        filename="srcadapt.dst")
  AT (0,0,0) RELATIVE a1 ROTATED (0,0,0) RELATIVE a1



/*
*/
```

B.3 SRCFLUX.INS

```
/*
*************************************************************************
 flux_source components of  INSTRUMENT PANDA
           - to be included into the main file panda.instr
             using %INCLUDE "srcflux.ins"
 usage: calculation of flux at different positions of the spectrometer

used parameters:
 mean_energy (meV) ...center of rectangular energy distr
 d_E         (mev) ... width/2 of rectangular energy distr
 src_flux (n/s/cm^2/AA)
*****************************************************************************/

/* this is the arm where the source is sitting on */
COMPONENT a1 = Arm()
  AT (0,0,0.3) ABSOLUTE

COMPONENT source = Source_flux(
        radius = 0.080, /* disc source at beginning of strahlrohr not too
big it has just the diameter 0.160 which covers  the strahlrohr
the next component is a diaphragm which makes only neutrons
pass from a rectangular area representing the strahlrohrnase */
        dist = 1.380,   /* distance to A2 ... entrance shutter*/
        xw = 0.105, yh = 0.098,
                                /* ENSRC     meV energy of source 1-6.3 A*/
        E0 = mean_energy,       /*  meV center of rectangular energy distr*/
        dE = d_E,               /*  mev  width/2 of rectangular energy
                                 distribution-difference to source_disc !!!! */
        flux= src_flux)  /* n/s/cm^2/AA */
  AT (0,0,0) RELATIVE a1 ROTATED (0,0,0) RELATIVE a1

/*strahlrohrnase*/
COMPONENT diasrc = Slit(
        xmin=-0.068,xmax=0.068,
        ymin = -0.0400, ymax = 0.0400)
  AT (0,0,0.002) RELATIVE a1  ROTATED (0,0,0) RELATIVE a1

/* flux monitor to detect the  total flux at the test position */
/*  COMPONENT fluxmon_test= Monitor_flux (                     */
/*    xmin=-0.005, xmax=0.005,         normal: =0.0525       */
/* ymin=-0.005, ymax=0.005,         normal: =0.054        */
/*      lmin=L_min,lmax=L_max)                                 */
/* AT (0,0,3.6) RELATIVE a1 ROTATED (0,0,0) RELATIVE a1        */


/*
*/
```

B.4 GUIDES1.INS

```
/*
*********************************************************
source to monochromator components
of  INSTRUMENT PANDA  - to be included into  main file panda.ins
using %INCLUDE "guides1.ins"
****************************************************************************/

COMPONENT guide1 = Guide(   /* fixed part of shutter */
w1=0.105,h1=0.098,
w2=0.105,h2=0.118,
l=1.325,
R0=0.75,   /*low angle reflectivity */
Qc=0.02,/* critical scattering vector */
alpha=6.49,  /* slope of reflectivity */
 /* slope of reflectivity evtl erniedrigen
                     (besserer supermirror als im manual*/
m=3,/*m value of material */
W=0.003333 ) /*1/300 ..width of supermirror cut off */
  AT (0, 0, 0) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2

 /*revolver guide*/
/* in cas m2 m3 m4 m5 there is no guid - thus we have only slits */
/*revolver 1*/
/*m1  COMPONENT guide2 = Guide(
w1=0.105,h1=0.118,
w2=0.105,h2=0.134,
l=1.015,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1  1m*/

/* 2 slits to simulate primary neutron path tube */
/*revolver 2*/
/*m2  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  2m*/
/*m4  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  4m*/
/*m5  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  5m*/
/* revolver 3*/
/*m3*/COMPONENT slit3 = Slit(
xmin=-0.020, xmax=0.020,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.069, ymax = 0.069)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3/*3m*/




COMPONENT lastwindow = Arm()
  AT (0,0,3.750) RELATIVE shutternase

/* no collimator in m1 mode !!*/
/*m1  COMPONENT guide3 = Guide(
w1=0.105,h1=0.134,
w2=0.105,h2=0.149,
l=1.000,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 2.340) RELATIVE a2  ROTATED (0,0,0) RELATIVE a2  1m*/

/*collimators in mode 2, 3,4,5 */
/*m2  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  2m*/
/*m3*/COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow/*3m*/
/*m4  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  4m*/
/*m5  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  5m*/


/*

*/
```

B.5 GUIDES2.INS

```
/*

***************************************************
 source to monochromator components WITHOUT VERTICAL SUPERMIRRORS IN GUIDE
of  INSTRUMENT PANDA  - to be included into the file src2mono.ins
using %INCLUDE "guides2.ins" just before the first guide
******************************************************************* */



COMPONENT slit1a1 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.049, ymax = 0.049)
  AT (0, 0, 0) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2

COMPONENT guide1 = Guide(
w1=0.105,h1=1.0,
w2=0.105,h2=1.0,
l=1.325,
R0=0.75,   /*low angle reflectivity */
```

```
Qc=0.02,/* critical scattering vector */
alpha=6.49,  /* slope of reflectivity */
 /* slope of reflectivity evtl erniedrigen
                      (besserer supermirror als im manual*/
m=3,/*m value of material */
W=0.003333 ) /*1/300 ..width of supermirror cut off */
  AT (0, 0, 0) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2

COMPONENT slit1a2 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2

 /*revolver guide*/
/* in cas m2 m3 there is no guid - thus we have only slits */
/*revolver 1*/
/*m1  COMPONENT slit2a1 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE guide1  ROTATED (0,0,0) RELATIVE guide1

        COMPONENT guide2 = Guide(
w1=0.105,h1=1.125,
w2=0.105,h2=1.145,
l=1.015,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1


COMPONENT slit2a2 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.067, ymax = 0.067)
  AT (0, 0, 1.015) RELATIVE guide2  ROTATED (0,0,0) RELATIVE guide2

  1m*/

/*revolver 2*/
/* 2 slits to simulate primary neutron path tube */
/*m2  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  2m*/
/*m4  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
```

```
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  4m*/
/*m5  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  5m*/
/*revolver 3*/
/*m3*/COMPONENT slit3 = Slit(
xmin=-0.020, xmax=0.020,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.069, ymax = 0.069)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3/*3m*/


COMPONENT lastwindow = Arm()
  AT (0,0,3.750) RELATIVE shutternase


/* no collimator in m1 mode !!*/
/*m1  COMPONENT slit3a1 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.067, ymax = 0.067)
  AT (0, 0, 2.340) RELATIVE a2  ROTATED (0,0,0) RELATIVE a2

COMPONENT guide3 = Guide(
w1=0.105,h1=1.0,
w2=0.105,h2=1.0,
l=1.000,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 2.340) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2


COMPONENT slit3a2 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.075, ymax = 0.075)
  AT (0, 0, 3.340) RELATIVE a2  ROTATED (0,0,0) RELATIVE a2
  1m*/


/*collimators in mode 2 and 3 ,4,5*/
/*m2  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  2m*/
/*m3*/COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow/*3m*/
/*m4  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  4m*/
/*m5  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  5m*/


/*

*/
```

B.6 GUIDES3.INS

```
/*

*******************************************************
source to monochromator components WITHOUT ANY SUPERMIRRORS IN GUIDE
of  INSTRUMENT PANDA  - to be included into the file panda.ins
using %INCLUDE "guides3.ins"
**************************************************************************/


COMPONENT slit1a1 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.049, ymax = 0.049)
  AT (0, 0, 0) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2

COMPONENT guide1 = Guide(
w1=1.0,h1=1.0,
w2=1.0,h2=1.0,
l=1.325,
R0=0.75,   /*low angle reflectivity */
Qc=0.02,/* critical scattering vector */
alpha=6.49,  /* slope of reflectivity */
 /* slope of reflectivity evtl erniedrigen
                      (besserer supermirror als im manual*/
m=3,/*m value of material */
W=0.003333 ) /*1/300 ..width of supermirror cut off */
  AT (0, 0, 0) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2

COMPONENT slit1a2 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2

 /*revolver guide*/
```

```
/* in cas m2 m3 there is no guid - thus we have only slits */
/*revolver 1*/
/*m1  COMPONENT slit2a1 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE guide1  ROTATED (0,0,0) RELATIVE guide1

        COMPONENT guide2 = Guide(
w1=1.0,h1=1.0,
w2=1.0,h2=1.0,
l=1.015,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1


COMPONENT slit2a2 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.067, ymax = 0.067)
  AT (0, 0, 1.015) RELATIVE guide2  ROTATED (0,0,0) RELATIVE guide2

  1m*/

/*revolver 2*/
/* 2 slits to simulate primary neutron path tube */
/*m2  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  2m*/
/*m4  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  4m*/
/*m5  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  5m*/
/*revolver 3*/
/*m3*/COMPONENT slit3 = Slit(
xmin=-0.020, xmax=0.020,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
```

```
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.069, ymax = 0.069)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3/*3m*/


COMPONENT lastwindow = Arm()
  AT (0,0,3.750) RELATIVE shutternase

/* no collimator in m1 mode !!*/
/*m1  COMPONENT slit3a1 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.067, ymax = 0.067)
  AT (0, 0, 2.340) RELATIVE a2  ROTATED (0,0,0) RELATIVE a2

COMPONENT guide3 = Guide(
w1=1.0,h1=1.0,
w2=1.0,h2=1.0,
l=1.000,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 2.340) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2


COMPONENT slit3a2 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.075, ymax = 0.075)
  AT (0, 0, 3.340) RELATIVE a2  ROTATED (0,0,0) RELATIVE a2
  1m*/



/*collimators in mode 2 and 3,4,5 */
/*m2  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  2m*/
/*m3*/COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow/*3m*/
/*m4  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  4m*/
/*m5  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  5m*/


/*

*/
```

B.7 GUIDES4.INS

```
/*

***************************************************
 source to monochromator components VERTICAL SUPERMIRRORS only in
 inner fix part of shutter
of  INSTRUMENT PANDA  - to be included into the file panda.ins
using %INCLUDE "guides4.ins"
******************************************************************* */


COMPONENT guide1 = Guide(
w1=0.105,h1=0.098,
w2=0.105,h2=0.118,
l=1.325,
R0=0.75,   /*low angle reflectivity */
Qc=0.02,/* critical scattering vector */
alpha=6.49,  /* slope of reflectivity */
 /* slope of reflectivity evtl erniedrigen
                      (besserer supermirror als im manual*/
m=3,/*m value of material */
W=0.003333 ) /*1/300 ..width of supermirror cut off */
  AT (0, 0, 0) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2

 /*revolver guide*/
/* in cas m2 m3 there is no guid - thus we have only slits */
/*revolver 1*/
/*m1  COMPONENT slit2a1 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE guide1  ROTATED (0,0,0) RELATIVE guide1

        COMPONENT guide2 = Guide(
w1=0.105,h1=1.0,
w2=0.105,h2=1.0,
l=1.015,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1


COMPONENT slit2a2 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.067, ymax = 0.067)
  AT (0, 0, 1.015) RELATIVE guide2  ROTATED (0,0,0) RELATIVE guide2
  1m*/

/* 2 slits to simulate primary neutron path tube */
/*revolver 2*/
/*m2  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  2m*/
/*m4  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  4m*/
/*m5  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  5m*/
/*revolver 3*/
/*m3*/COMPONENT slit3 = Slit(
xmin=-0.020, xmax=0.020,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.069, ymax = 0.069)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3/*3m*/


COMPONENT lastwindow = Arm()
  AT (0,0,3.750) RELATIVE shutternase

/* no collimator in m1 mode !!*/
/*m1  COMPONENT slit3a1 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.067, ymax = 0.067)
  AT (0, 0, 2.340) RELATIVE a2  ROTATED (0,0,0) RELATIVE a2

COMPONENT guide3 = Guide(
w1=0.105,h1=1.0,
w2=0.105,h2=1.0,
l=1.000,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 2.340) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2


COMPONENT slit3a2 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.075, ymax = 0.075)
  AT (0, 0, 3.340) RELATIVE a2  ROTATED (0,0,0) RELATIVE a2
  1m*/


/*collimators in mode 2 and 3,4,5 */
/*m2  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  2m*/
/*m3*/COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
  AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow/*3m*/
/*m4  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
  AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  4m*/
/*m5  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
  AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  5m*/


/*
*/
```

B.8 GUIDES5.INS

```
/*
***************************************************
 source to monochromator components WITH VERTICAL SUPERMIRRORS only
 in inner fix part and revolver GUIDE of  INSTRUMENT PANDA
  - to be included into the file panda.ins
using %INCLUDE "guides5.ins"
******************************************************************* */


COMPONENT guide1 = Guide(
w1=0.105,h1=0.098,
w2=0.105,h2=0.118,
l=1.325,
R0=0.75,   /*low angle reflectivity */
Qc=0.02,/* critical scattering vector */
alpha=6.49,  /* slope of reflectivity */
 /* slope of reflectivity evtl erniedrigen
                      (besserer supermirror als im manual*/
m=3,/*m value of material */
W=0.003333 ) /*1/300 ..width of supermirror cut off */
  AT (0, 0, 0) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2

 /*revolver guide*/
/* in cas m2 m3 there is no guid - thus we have only slits */
/*revolver 1*/
/*m1  COMPONENT guide2 = Guide(
w1=0.105,h1=0.118,
w2=0.105,h2=0.134,
l=1.015,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1

  1m*/

/* 2 slits to simulate primary neutron path tube */
/*revolver 2*/
/*m2  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  2m*/
/*m4  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  4m*/
/*m5  COMPONENT slit3 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.050, ymax = 0.050)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.053, ymax = 0.053)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3  5m*/
/*revolver 3*/
/*m3*/COMPONENT slit3 = Slit(
xmin=-0.020, xmax=0.020,
ymin = -0.059, ymax = 0.059)
  AT (0, 0, 1.325) RELATIVE guide1 ROTATED (0,0,0) RELATIVE guide1
COMPONENT slit4 = Slit(
xmin=-0.025, xmax=0.025,
ymin = -0.069, ymax = 0.069)
  AT (0, 0, 1.015) RELATIVE slit3 ROTATED (0,0,0) RELATIVE slit3/*3m*/


COMPONENT lastwindow = Arm()
  AT (0,0,3.750) RELATIVE shutternase

/* no collimator in m1 mode !!*/
/*m1  COMPONENT slit3a1 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.067, ymax = 0.067)
  AT (0, 0, 2.340) RELATIVE a2  ROTATED (0,0,0) RELATIVE a2

COMPONENT guide3 = Guide(
w1=0.105,h1=1.0,
w2=0.105,h2=1.0,
l=1.000,
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333 )
  AT (0, 0, 2.340) RELATIVE a2 ROTATED (0,0,0) RELATIVE a2


COMPONENT slit3a2 = Slit(
xmin=-0.0525, xmax=0.0525,
ymin = -0.075, ymax = 0.075)
  AT (0, 0, 3.340) RELATIVE a2  ROTATED (0,0,0) RELATIVE a2
  1m*/


/*collimators in mode 2 and 3,4,5 */
/*m2  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  2m*/
/*m3*/COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91)
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow/*3m*/
/*m4  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91 )
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  4m*/
/*m5  COMPONENT Coll1= Soller_trans(
        xmin=-0.025,xmax=0.025,
ymin=-0.080,ymax=0.080,
len=0.500,
divergence=40, transmission=0.91 )
```

```
 AT (0,0,0.30) RELATIVE lastwindow ROTATED (0,0,0) RELATIVE lastwindow  5m*/


/*
*/
```

B.9 MONDET.INS

```
/*
***********************************************************************
guide to monochromator components of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "mondet.ins"

parameters:
 A1(deg)              theta monochromator
 ninetyminusA1
 minusA2 A2.........  2theta monochromator
 L_min,L_max (A)      wavelength interval for detectors
 E_min,E_max (meV)    energy interval for detectors

********************************************************************************/


/* Divergence Monitor to detect the divergence of the beam profile referring to
a plane normal to neutron path before the  monochromator */

COMPONENT divmon_mono= Divergence_monitor (
   xmin=-0.115, xmax=0.115,
ymin=-0.100, ymax=0.100,
nh=50,nv=50,
v_maxdiv=5,h_maxdiv=5,
filename="div.mon")
 AT (0,0,7.6) RELATIVE shutternase  ROTATED (0,0,0) RELATIVE shutternase


/* PSD monitor to detect the incoming beam profile on the monochromator */
COMPONENT psd_monoprofile= PSD_monitor (
    xmin=-0.115, xmax=0.115,
ymin=-0.100, ymax=0.100,
nx=50,ny=50,
filename="prof.mon")
  AT (0,0,7.879) RELATIVE shutternase  ROTATED (0,ninetyminusA1,0) RELATIVE shutternase



/* energy monitor to detect the  energy profile at the monochromator*/
COMPONENT emon_mono= E_monitor (
    xmin=-0.115, xmax=0.115,
ymin=-0.100, ymax=0.100,
Emin=E_min,Emax=E_max,nchan=80,
filename="emon.mon")
  AT (0,0,7.879) RELATIVE shutternase ROTATED (0,ninetyminusA1,0) RELATIVE shutternase

/* flux monitor to detect the  total flux at the monochromator*/
```

```
COMPONENT fluxmon_mono= Monitor_flux (
    xmin=-0.115, xmax=0.115,
ymin=-0.100, ymax=0.100,
lmin=L_min,lmax=L_max)
  AT (0,0,7.879) RELATIVE shutternase ROTATED (0,ninetyminusA1,0) RELATIVE shutternase


/*

*/
```

B.10 CUMON.INS

```
/*

************************************************************
Copper Monochromator of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "cumon.ins"

parameters:
r0_mono, maximum reflectivity (1)
        Q_mono,   qvector(1/AA)
       R_V,R_H  radius of vertical/horizontal focussing
************************************************************/


COMPONENT cu_mono = Mon_2foc(
        zwidth=0.017, /*horizontal (width) of an individual slab */
        ywidth=0.020, /*vertical   (height) of an individual slab */
gap=0.0005,  /*typical gap between adjacent slabs */
NH=11,NV=11,  /*number of slabs horizontal (columns), vertical (rows) */

mosaich=24, mosaicv=6, /* horizontal, vertical mosaic FWHM (arc minutes)*/
r0=r0_mono, /*maximum reflectivity (1) */
        Q=Q_mono,  /* qvector(1/AA) for 111  */
       RV=R_V,RH=R_H /* radius of vertical/horizontal focussing */
)
  AT (0,0,7.880) RELATIVE shutternase ROTATED (0,minusA1,0) RELATIVE shutternase


 /*

*/
```

B.11 GRAPHMON.INS

```
/*

************************************************************
Graphit Monochromator of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "graphmon.ins"

parameters:
r0_mono, maximum reflectivity (1)
        Q_mono,   qvector(1/AA)
       R_V,R_H  radius of vertical/horizontal focussing
************************************************************/
```

```
COMPONENT graphit_mono = Mon_2foc(
    zwidth=0.018, /*horizontal (width) of an individual slab */
    ywidth=0.020, /*vertical   (height) of an individual slab */
    gap=0.0005,  /*typical gap between adjacent slabs */
    NH=11,NV=11,  /*number of slabs horizontal (columns), vertical (rows) */
    mosaich=30, mosaicv=30, /* horizontal, vertical mosaic FWHM (arc minutes)*/
    r0=r0_mono, /*maximum reflectivity (1) */
    Q=Q_mono,  /* qvector(1/AA) */
    RV=R_V,RH=R_H /* radius of vertical/horizontal focussing */
)
AT (0,0,7.880) RELATIVE shutternase ROTATED (0,minusA1,0) RELATIVE shutternase


/*
*/
```

B.12 HSMON.INS

```
/*
************************************************************
Heusler Monochromator of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "hsmon.ins"

parameters:
r0_mono, maximum reflectivity (1)
        Q_mono,   qvector(1/AA)
       R_V,R_H  radius of vertical/horizontal focussing
************************************************************/


COMPONENT heusler_mono = Mon_2foc(
    zwidth=0.140, /*horizontal (width) of an individual slab */
    ywidth=0.015, /*vertical   (height) of an individual slab */
    gap=0.0005,  /*typical gap between adjacent slabs */
    NH=1,NV=9,  /*number of slabs horizontal (columns), vertical (rows) */
    mosaich=??, mosaicv=??, /* horizontal, vertical mosaic FWHM (arc minutes)*/
    r0=r0_mono, /*maximum reflectivity (1) */
    Q=Q_mono,  /* qvector(1/AA) */
    RV=R_V,RH=0 /* radius of vertical/horizontal focussing */
)
AT (0,0,7.880) RELATIVE shutternase ROTATED (0,minusA1,0) RELATIVE shutternase


/*
*/
```

B.13 MON2SPLE.INS

```
/*
*******************************************************
monochromator to sample components
of  INSTRUMENT PANDA  - to be included into the file panda.ins
```

```
using %INCLUDE "mon2sple.ins" just after the monochromator
**************************************************************************/

/* convergent neutron guide for mode 2 */
/*m2  COMPONENT guideconv = Guide(
w1=0.105,h1=0.098,  ???
w2=0.105,h2=0.118,  ???
l=1.325,  ???
R0=0.75,
Qc=0.02,
alpha=6.49,
m=3,
W=0.003333)
  AT (0, 0, ??) RELATIVE A2arm ROTATED (0,0,0) RELATIVE A2arm
2m*/


/*collimator after monochromator for mode  3,4,5 */
/*m3*/COMPONENT coll2 = Soller_trans(
         xmin=-0.020, xmax= 0.020,
 ymin=-0.060, ymax= 0.060,
 len=0.20,
 divergence=40, transmission=0.97 )
  AT (0, 0, 1.000) RELATIVE A2arm  ROTATED (0,0,0) RELATIVE A2arm
/*3m*/
/*m4  COMPONENT coll2 = Soller_trans(
         xmin=-0.020, xmax= 0.020,
 ymin=-0.060, ymax= 0.060,
 len=0.20,
 divergence=40, transmission=0.97)
  AT (0, 0, 1.000) RELATIVE A2arm  ROTATED (0,0,0) RELATIVE A2arm
  4m*/
/*m5    COMPONENT coll2 = Soller_trans(
         xmin=-0.020, xmax= 0.020,
 ymin=-0.060, ymax= 0.060,
 len=0.20,
 divergence=40, transmission=0.97)
  AT (0, 0, 1.000) RELATIVE A2arm  ROTATED (0,0,0) RELATIVE A2arm
  5m*/

/*

*/
```

B.14 SPLFLDET.INS

```
/*

**********************************************************************
sample flux detectors of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "splfldet.ins"

parameters:
 L_min,L_max (A)      wavelength interval for detectors
 E_min,E_max (meV)    energy interval for detectors

*******************************************************************************/
```

```
/* Divergence monitor at sample position*/
COMPONENT divmon_sample= Divergence_monitor (
   xmin=-0.025, xmax=0.025,
ymin=-0.025, ymax=0.025,
nh=50,nv=50,
v_maxdiv=5,h_maxdiv=5,
filename="div.sam")
  AT (0,0,1.994) RELATIVE A2arm ROTATED (0,0,0) RELATIVE A2arm


/* PSD monitor to detect the incoming beam profile on the sample position */
COMPONENT psd_sampleprofile= PSD_monitor (
    xmin=-0.025, xmax=0.025,
ymin=-0.025, ymax=0.025,
nx=25,ny=25,
filename="prof.sam")
  AT (0,0,1.994) RELATIVE A2arm ROTATED (0,0,0) RELATIVE A2arm



/* energy monitor to detect the  energy profile at the sample position*/
COMPONENT emon_sample= E_monitor (
    xmin=-0.025, xmax=0.025,
ymin=-0.025, ymax=0.025,
Emin=E_min,Emax=E_max,nchan=80,
filename="emon.sam")
  AT (0,0,1.994) RELATIVE A2arm ROTATED (0,0,0) RELATIVE A2arm

/* flux monitor to detect the  total flux at the sample position*/
COMPONENT fluxmon_sample= Monitor_flux (
    xmin=-0.025, xmax=0.025,
ymin=-0.025, ymax=0.025,
lmin=L_min,lmax=L_max)
  AT (0,0,1.994) RELATIVE A2arm ROTATED (0,0,0) RELATIVE A2arm


/* 1x1 Divergence monitor at sample position*/
COMPONENT divmon_1x1sample= Divergence_monitor (
   xmin=-0.005, xmax=0.005,
ymin=-0.005, ymax=0.005,
nh=5,nv=5,
v_maxdiv=5,h_maxdiv=5,
filename="div1x1.sam")
  AT (0,0,1.994) RELATIVE A2arm ROTATED (0,0,0) RELATIVE A2arm




/* 0.01x0.01 PSD monitor to detect the incoming beam profile on the 1cm x1cm sample */
COMPONENT psd_1x1sampleprofile= PSD_monitor (
    xmin=-0.005, xmax=0.005,
ymin=-0.005, ymax=0.005,
nx=5,ny=5,
filename="prof1x1.sam")
  AT (0,0,1.994) RELATIVE A2arm ROTATED (0,0,0) RELATIVE A2arm
```

```
/* 0.01x0.01 energy monitor to detect the  energy profile at the 1cmx1cm sample*/
COMPONENT emon_1x1sample= E_monitor (
    xmin=-0.005, xmax=0.005,
ymin=-0.005, ymax=0.005,
Emin=E_min,Emax=E_max,nchan=16,
filename="emon1x1.sam")
  AT (0,0,1.994) RELATIVE A2arm ROTATED (0,0,0) RELATIVE A2arm


/* 0.01x0.01 flux monitor to detect the  total flux at 1cmx1cm  sample*/
COMPONENT fluxmon_1x1sample= Monitor_flux (
    xmin=-0.005, xmax=0.005,
ymin=-0.005, ymax=0.005,
lmin=L_min,lmax=L_max)
  AT (0,0,1.994) RELATIVE A2arm ROTATED (0,0,0) RELATIVE A2arm



/*

*/
```

B.15 VANSPL.INS

```
/*

************************************************************
VANADIUM sample of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "VANSPL.INS"

usage: calculation of the resolution function

parameters:
 dis_sample_analyzer (m)
 A3    .... 2theta sample
*****************************************************************************/



COMPONENT sample= V_sample (
                radius_i=0.003,  /*inner radius of cylinder(m)*/
                radius_o=0.005, /*outer radius of cylinder(m)*/
                h=0.02,   /*sample height (m)*/
                pack=1, /*packing factor (1)*/
target_x=targetx,  /*target location and sizer (radius)*/
                target_y=0,
                target_z=targetz,
                focus_r=0.1)/*corresponds to analyzer size*/
  AT (0,0,2.000) RELATIVE A2arm ROTATED (0,A3,0) RELATIVE A2arm


/*

*/
```

B.16 RESSPL.INS

```
/*

************************************************************
uniform scattering sample of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "resspl.ins"

usage: calculation of the resolution function

parameters:
 dis_sample_analyzer (m)
 E_f     .... final energy (meV)
 d_E    .... energy bandwidth of scattering
 A4     .... 2theta sample
*****************************************************************************/



COMPONENT sample= Res_sample (
                radius_i=0,  /*inner radius of cylinder(m)*/
                radius_o=0.005, /*outer radius of cylinder(m)*/
                h=0.02,   /*sample height (m)*/
                target_x=targetx,  /*target location and sizer (radius)*/
                target_y=0,
                target_z=targetz,
                focus_r=0.1,/*corresponds to analyzer size*/
                E0=E_f,  /*centre of energy spread (meV)= Efinal*/
                dE=d_E) /*energy spread (meV) -matching src and detectors*/
  AT (0,0,2.000) RELATIVE A2arm ROTATED (0,A3,0) RELATIVE A2arm


/*
*/
```

B.17 SPL2ANA.INS

```
/*

**********************************************************************
Analyzer  of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "spl2ana.ins"

usage: calculation of the resolution function

parameters:
 dist_sample_analyzer (m)
 dist_sample_analyzer_screening_outside (m)
 dist_sample_dia (m)
 dist_sample_mon (m)
 E_f    .... final energy (meV)
 d_E    .... energy bandwidth of scattering
 A4     .... 2theta sample
*******************************************************************************/

COMPONENT sample_analyzer_arm=Arm()
```

```
  AT (0,0,2.000) RELATIVE A2arm ROTATED (0,A4,0) RELATIVE A2arm

/*entrance channel of analyzer - outer end*/
COMPONENT dia_an_in1= Slit(
         xmin=-0.030, xmax=0.030,
 ymin= -0.070,ymax=0.070)
 AT (0,0,dist_sample_analyzer_screening_outside) RELATIVE sample_analyzer_arm
          ROTATED (0,0,0) RELATIVE sample_analyzer_arm


/*collimator 3 in mode 3 and 4*/
/*m3*/COMPONENT Coll3= Soller_trans(
        xmin=-0.020,xmax=0.020,
ymin=-0.060,ymax=0.060,
len=0.200,
divergence=40, transmission=0.97)
 AT (0,0,dist_sample_analyzer_screening_outside) RELATIVE sample_analyzer_arm
         ROTATED (0,0,0) RELATIVE sample_analyzer_arm/*3m*/
/*m4  COMPONENT Coll3= Soller_trans(
        xmin=-0.020,xmax=0.020,
ymin=-0.060,ymax=0.060,
len=0.200,
divergence=40, transmission=0.97)
 AT (0,0,dist_sample_analyzer_screening_outside) RELATIVE sample_analyzer_arm
         ROTATED (0,0,0) RELATIVE sample_analyzer_arm  4m*/


/*entrance channel of analyzer - inner end */
COMPONENT dia_an_in2 = Slit(
        xmin=-0.035,xmax=0.035,
        ymin = -0.060, ymax = 0.060)
  AT (0,0,dist_sample_dia) RELATIVE sample_analyzer_arm
           ROTATED (0,0,0) RELATIVE sample_analyzer_arm


/* additional diaphragm in analyzer */
COMPONENT dia_analyser_in = Slit(
        xmin=-0.025,xmax=0.025,
        ymin = -0.060, ymax = 0.060) /* genaue Breite u Hoehe  muss erst festgelegt werden*/
 AT (0,0,dist_sample_dia) RELATIVE sample_analyzer_arm
          ROTATED (0,0,0) RELATIVE sample_analyzer_arm


/*resolution ellipsoid monitor - at this position there is really a monitor*/
/*COMPONENT resmonitor_ell=Res_monitor (
            xmin=-0.020,xmax=0.020,
            ymin=-0.06,ymax=0.06,
            filename="resmon.ell",
            res_sample_comp=sample)
AT (0,0,dist_sample_mon) RELATIVE sample_analyzer_arm
         ROTATED (0,0,0) RELATIVE sample_analyzer_arm*/


/* Divergence monitor at analyzer  position*/
COMPONENT divmon_ana= Divergence_monitor (
                xmin=-0.085, xmax=0.085,
                ymin=-0.060, ymax=0.060,
nh=50,nv=50,
v_maxdiv=5,h_maxdiv=5,
filename="div.ana")
  AT (0,0,dist_sample_analyzer_10) RELATIVE sample_analyzer_arm
      ROTATED (0,0,0) RELATIVE sample_analyzer_arm



/* energy monitor to detect the  energy profile at the analyser*/
COMPONENT emon_ana= E_monitor (
        xmin=-0.085, xmax=0.085,
        ymin=-0.060, ymax=0.060,
        Emin=E_min,Emax=E_max,nchan=40,
        filename="emon.ana")
  AT (0,0,dist_sample_analyzer_1) RELATIVE sample_analyzer_arm
      ROTATED (0,ninetyminusA5,0) RELATIVE sample_analyzer_arm



/* PSD monitor to detect the incoming beam profile on the analyser */
COMPONENT psd_anaprofile= PSD_monitor (
        xmin=-0.085, xmax=0.085,
        ymin=-0.060, ymax=0.060,
        nx=50,ny=50,
        filename="prof.ana")
  AT (0,0,dist_sample_analyzer_1) RELATIVE sample_analyzer_arm
      ROTATED (0,ninetyminusA5,0) RELATIVE sample_analyzer_arm





/* flux monitor to detect the  total flux at the analyser*/

COMPONENT fluxmon_ana= Monitor_flux (
        xmin=-0.085, xmax=0.085,
        ymin=-0.060, ymax=0.060,
        lmin=L_min,lmax=L_max)
  AT (0,0,dist_sample_analyzer_1) RELATIVE sample_analyzer_arm
      ROTATED (0,ninetyminusA5,0) RELATIVE sample_analyzer_arm
```

/*

*/T

B.18 GRAPHANA.INS

/*

```
************************************************************
Graphit Analyzer of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
```

```
     using %INCLUDE "graphana.ins"

parameters:
r0_ana, maximum reflectivity (1)
        Q_ana,   qvector(1/AA)
       R_Vana,R_Hana  radius of vertical/horizontal focussing
distance_sample_analyzer (m)
************************************************************/

/* graphit analyzer for PANDA.instr */
COMPONENT analyzer = Mon_2foc(
        zwidth=0.015, /*horizontal (width) of an individual slab */
        ywidth=0.030, /*vertical   (height) of an individual slab */
        gap=0.0005,  /*typical gap between adjacent slabs */
        NH=11,NV=5,  /*number of slabs horizontal (columns), vertical (rows)*/
        mosaich=30, mosaicv=30, /* horizontal, vertical mosaic FWHM (arc minute*/
        r0=r0_ana, /*maximum reflectivity (1) */
        Q=Q_ana,  /* qvector(1/AA) */
        RV=R_Vana,RH=R_Hana /* radius of vertical/horizontal focussing */
        )
AT (0,0,dist_sample_analyzer) RELATIVE sample_analyzer_arm
                      ROTATED (0,minusA5,0) RELATIVE sample_analyzer_arm



/*

*/
```

B.19 CUANA.INS

```
/*

************************************************************
Copper Analyzer of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "cuana.ins"

parameters:
r0_ana, maximum reflectivity (1)
        Q_ana,   qvector(1/AA)
       R_Vana,R_Hana  radius of vertical/horizontal focussing
distance_sample_analyzer (m)
************************************************************/

/* copper analyzer for PANDA.instr */
COMPONENT analyzer = Mon_2foc(
        zwidth=0.015, /*horizontal (width) of an individual slab */
        ywidth=0.030, /*vertical   (height) of an individual slab */
        gap=0.0005,  /*typical gap between adjacent slabs */
        NH=11,NV=5,  /*number of slabs horizontal (columns), vertical (rows)*/
        mosaich=24, mosaicv=6, /* horizontal, vertical mosaic FWHM (arc minute*/
        r0=r0_ana, /*maximum reflectivity (1) */
        Q=Q_ana,  /* qvector(1/AA) */
        RV=R_Vana,RH=R_Hana /* radius of vertical/horizontal focussing */
        )
AT (0,0,dist_sample_analyzer) RELATIVE sample_analyzer_arm
                      ROTATED (0,minusA5,0) RELATIVE sample_analyzer_arm
```

```
/*

*/
```

B.20 HSANA.INS

```
/*

************************************************************
Heusler Analyzer of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "hsana.ins"

parameters:
r0_ana, maximum reflectivity (1)
        Q_ana,   qvector(1/AA)
       R_Vana,R_Hana  radius of vertical/horizontal focussing
distance_sample_analyzer (m)
************************************************************/

/* Heusler analyzer for PANDA.instr */
COMPONENT analyzer = Mon_2foc(
        zwidth=0.015, /*horizontal (width) of an individual slab */
        ywidth=0.025, /*vertical   (height) of an individual slab */
        gap=0.0005,  /*typical gap between adjacent slabs */
        NH=11,NV=5,  /*number of slabs horizontal (columns), vertical (rows)*/
        mosaich=??, mosaicv=??, /* horizontal, vertical mosaic FWHM (arc minute*/
        r0=r0_ana, /*maximum reflectivity (1) */
        Q=Q_ana,  /* qvector(1/AA) */
        RV=0.543 ,RH=R_Hana /* radius of vertical/horizontal focussing  vertical
                                                                                    fixed to 0.543m=0.04
        )
AT (0,0,dist_sample_analyzer) RELATIVE sample_analyzer_arm
                      ROTATED (0,minusA5,0) RELATIVE sample_analyzer_arm



/*

*/
```

B.21 DETECTOR.INS

```
/*

***********************************************************
 detectors of  INSTRUMENT PANDA
   - to be included into the main file panda.ins
     using %INCLUDE "detector.ins"

parameters:
 L_min,L_max (A)      wavelength interval for detectors
 E_min,E_max (meV)    energy interval for detectors
 dist_sample_analyzer (m)
 dist_analyzer_detector (m)
*******************************************************************************/
```

```
COMPONENT analyzer_detector_arm=Arm()
AT  (0,0,dist_sample_analyzer) RELATIVE sample_analyzer_arm ROTATED
(0,minusA6,0) RELATIVE sample_analyzer_arm

/* collimator before detector in collimated modes 3 and 4 */
/*m3*/COMPONENT coll4 = Soller_trans(
         xmin=-0.020, xmax= 0.020,
 ymin=-0.060, ymax= 0.060,
 len=0.20,
 divergence=40, transmission=0.97)
  AT (0, 0, dist_analyzer_coll4) RELATIVE analyzer_detector_arm
     ROTATED (0,0,0) RELATIVE analyzer_detector_arm
/*3m*/
/*m4  COMPONENT coll4 = Soller_trans(
         xmin=-0.020, xmax= 0.020,
 ymin=-0.060, ymax= 0.060,
 len=0.20,
 divergence=40, transmission=0.97)
  AT (0, 0, dist_analyzer_coll4) RELATIVE analyzer_detector_arm
     ROTATED (0,0,0) RELATIVE analyzer_detector_arm
  4m*/

/***************DETECTORS TO DETERMINE BEAM PROFILE*************/


/* Divergence monitor at detecotor position*/
COMPONENT divmon_det= Divergence_monitor (
   xmin=-0.025, xmax=0.025,
ymin=-0.060, ymax=0.060,
nh=50,nv=50,
v_maxdiv=5,h_maxdiv=5,
filename="div.det")
  AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                               ROTATED (0,0,0) RELATIVE analyzer_detector_arm



/* energy monitor to detect the  energy profile */
COMPONENT eprofile_det= E_monitor (
    xmin=-0.025, xmax=0.025,
ymin=-0.060, ymax=0.060,
Emin=E_min,Emax=E_max,nchan=40,
filename="emon.det")
  AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                               ROTATED (0,0,0) RELATIVE analyzer_detector_arm

/* PSD monitor to detect the incoming beam profile  */
COMPONENT psdprofile_det= PSD_monitor (
    xmin=-0.0250, xmax=0.025,
ymin=-0.060, ymax=0.060,
nx=25,ny=25,
filename="prof.det")
  AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                               ROTATED (0,0,0) RELATIVE analyzer_detector_arm
```

```
/********REAL DETECTORS ****************************/

/* for focussing mode (1,2,5) detector is only 14mm diamter - to simulate this
we put in front of the detectors a diaphragm */
/*m1  COMPONENT slit_det = Slit(xmin=-0.0075,xmax=0.0075,ymin=-0.060,ymax=0.060)
 AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                 ROTATED (0,0,0) RELATIVE analyzer_detector_arm  1m*/
/*m2  COMPONENT slit_det = Slit(xmin=-0.0075,xmax=0.0075,ymin=-0.060,ymax=0.060)
 AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                 ROTATED (0,0,0) RELATIVE analyzer_detector_arm  2m*/
/*m5  COMPONENT slit_det = Slit(xmin=-0.0075,xmax=0.0075,ymin=-0.060,ymax=0.060)
 AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                 ROTATED (0,0,0) RELATIVE analyzer_detector_arm  5m*/


/* this is to calculate which neutrons are 'good' */
COMPONENT adaptor=Adapt_check (source_comp=source)
  AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                               ROTATED (0,0,0) RELATIVE analyzer_detector_arm




/* Divergence monitor at detecotor position*/
COMPONENT divmon_realdet= Divergence_monitor (
   xmin=-0.0125, xmax=0.0125,
ymin=-0.060, ymax=0.060,
nh=50,nv=50,
v_maxdiv=5,h_maxdiv=5,
filename="realdiv.det")
  AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                               ROTATED (0,0,0) RELATIVE analyzer_detector_arm



/* energy monitor to detect the  energy profile */
COMPONENT eprofile_realdet= E_monitor (
    xmin=-0.0125, xmax=0.0125,
ymin=-0.060, ymax=0.060,
Emin=E_min,Emax=E_max,nchan=40,
filename="realemon.det")
  AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                               ROTATED (0,0,0) RELATIVE analyzer_detector_arm

/* PSD monitor to detect the incoming beam profile  */
COMPONENT psdprofile_realdet= PSD_monitor (
    xmin=-0.01250, xmax=0.0125,
ymin=-0.060, ymax=0.060,
nx=25,ny=25,
filename="realprof.det")
  AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                               ROTATED (0,0,0) RELATIVE analyzer_detector_arm
```

```
/* flux monitor to detect the  total flux */
COMPONENT fluxmon_detektor= Monitor_flux (
    xmin=-0.0125, xmax=0.0125,
ymin=-0.060, ymax=0.060,
lmin=L_min,lmax=L_max)
  AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
                               ROTATED (0,0,0) RELATIVE analyzer_detector_arm


/*resolution ellipsoid monitor*/
/*ressplonly*/COMPONENT res_detektor=Res_monitor (
    xmin=-0.0125, xmax=0.0125,
ymin=-0.060, ymax=0.060,
            filename="resmon.inel",
            res_sample_comp=sample)
  AT (0,0,dist_analyzer_detector) RELATIVE analyzer_detector_arm
  ROTATED (0,0,0) RELATIVE analyzer_detector_arm/*onlyresspl*/


/*

*/
```