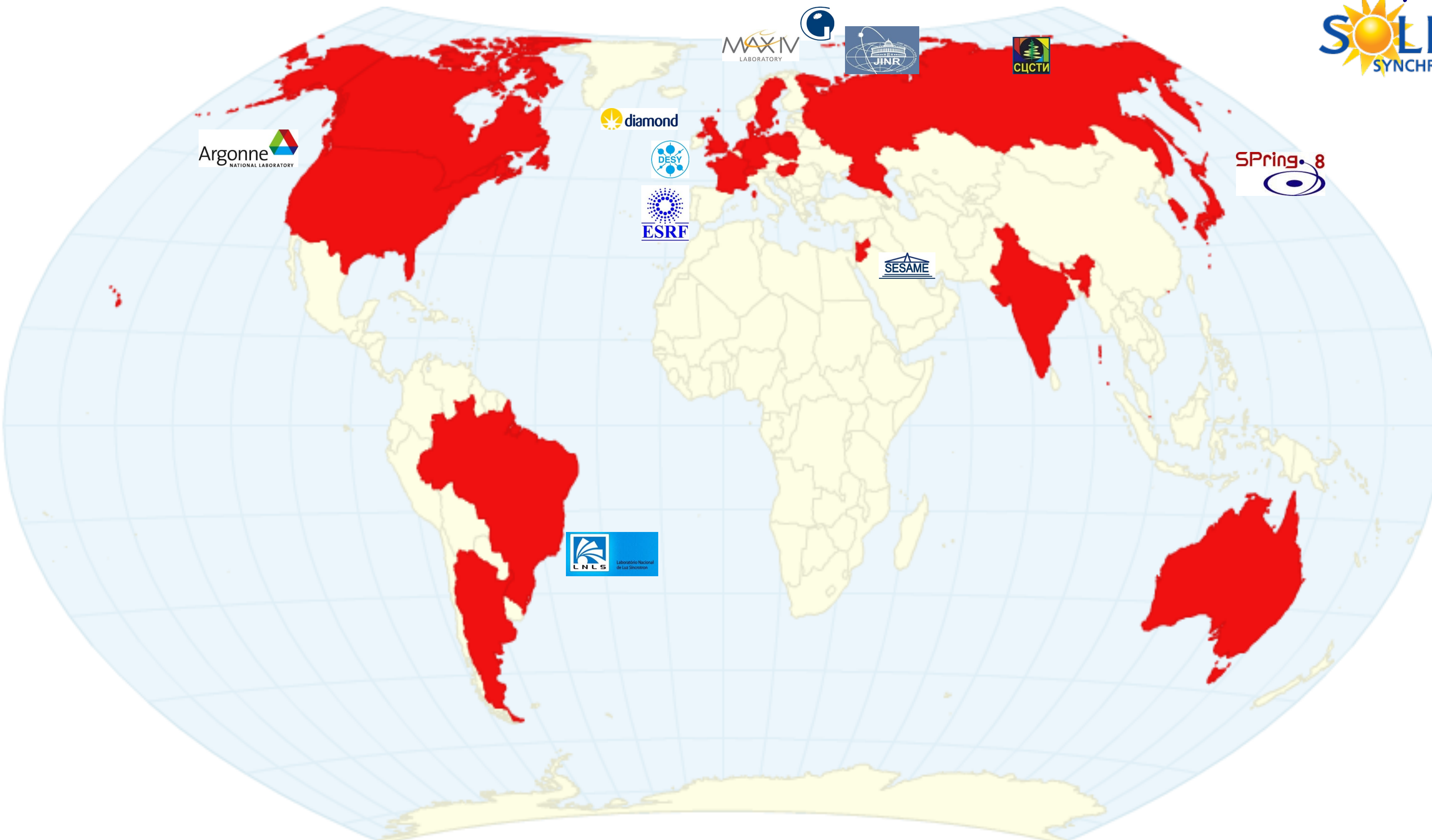
McXtrace

Powered by McStas technology

"McXtrace: a Monte Carlo software package for simulating X-ray optics, beamlines and experiments",
Journal of Applied Crystallography, vol. 46, part 3, June 2013

Peter Willendrup <pkwi@fysik.dtu.dk>
(slides from Erik Bergbäck Knudsen)

Where?



MAX IV LABORATORY
JINR
СЦСТИ
diamond
Argonne NATIONAL LABORATORY
DESY
ESRF
SESAME
SPring-8
LNLS

What is it?

- Built on proven base of McStas for neutron ray tracing
  K. Lefmann and K. Nielsen, Neutron News 10, 20, (1999).
- Releases 1.7 and 3.0 (GPU support)
- Portable code (Unix/Linux/Mac/Windows, 32 and 64 bit support)
  Has run on all from iPhone to FERMI (world no. 7)

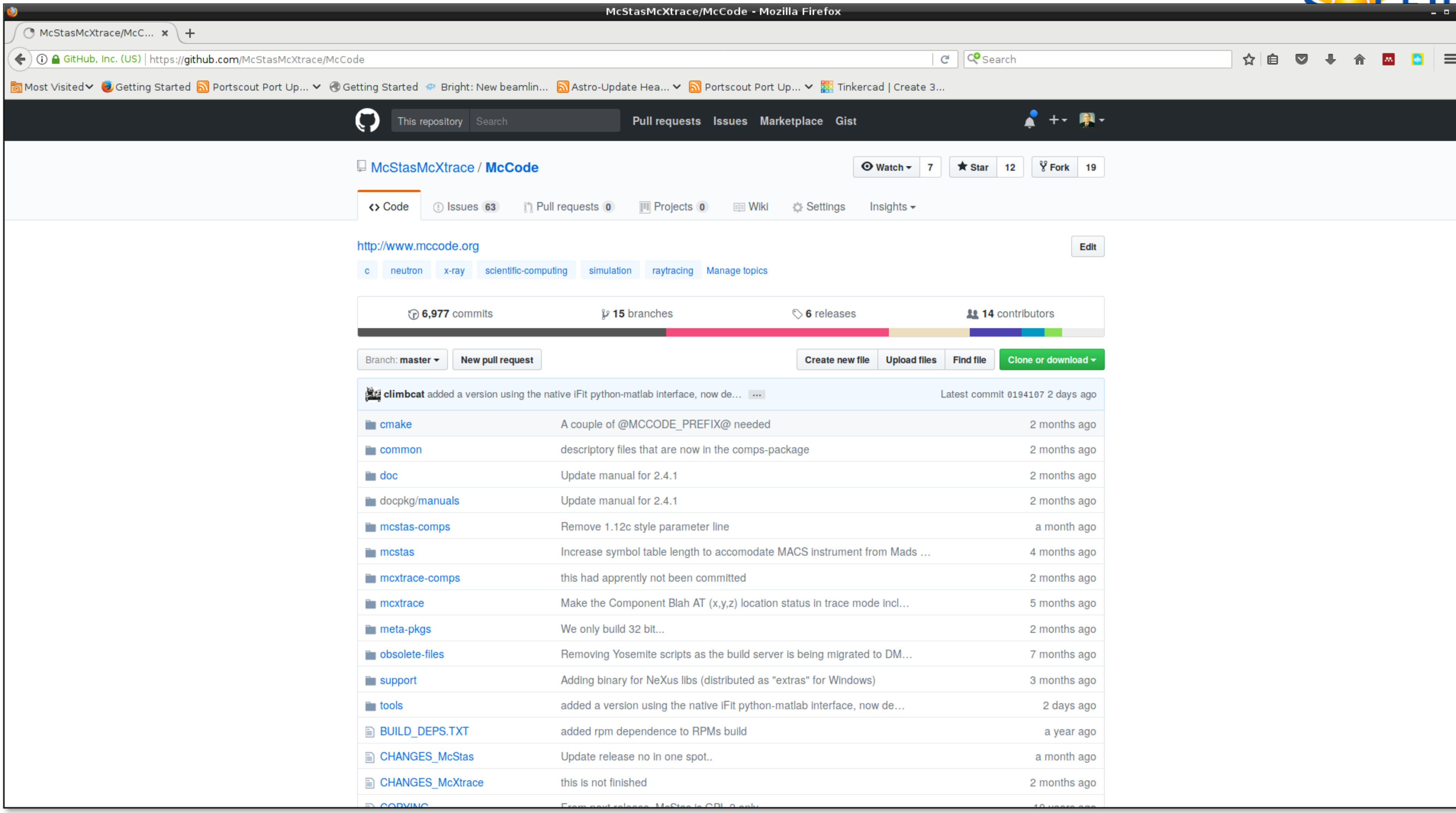



Project website at
http://www.mcxtrace.org

Project mailing list at
mcxtrace-users@mcxtrace.org

- GPL-license
- DSL / Compiler Technology.
  Using Lex & Yacc
- Modular Open Structure.
  Components/devices written in structured ISO-c automatically fits in the system
- Dependencies: c-compiler (python3/qt5 or perl/tk for gui).
- Permanent staff at DTU Physics maintaining the code

Github



McStasMcXtrace/McCode - Mozilla Firefox
https://github.com/McStasMcXtrace/McCode
McStasMcXtrace / McCode
Watch 7
Star 12
Fork 19
Code
Issues 63
Pull requests 0
Projects 0
Wiki
Settings
Insights
http://www.mccode.org
c
neutron
x-ray
scientific-computing
simulation
raytracing
Manage topics
6,977 commits
15 branches
6 releases
14 contributors
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
climbcat added a version using the native iFit python-matlab interface, now de…
Latest commit 0194107 2 days ago
cmake
A couple of @MCCODE_PREFIX@ needed
2 months ago
common
descriptory files that are now in the comps-package
2 months ago
doc
Update manual for 2.4.1
2 months ago
docpkg/manuals
Update manual for 2.4.1
2 months ago
mcstas-comps
Remove 1.12c style parameter line
a month ago
mcstas
Increase symbol table length to accomodate MACS instrument from Mads …
4 months ago
mcxtrace-comps
this had apprently not been committed
2 months ago
mcxtrace
Make the Component Blah AT (x,y,z) location status in trace mode incl…
5 months ago
meta-pkgs
We only build 32 bit...
2 months ago
obsolete-files
Removing Yosemite scripts as the build server is being migrated to DM…
7 months ago
support
Adding binary for NeXus libs (distributed as "extras" for Windows)
3 months ago
tools
added a version using the native iFit python-matlab interface, now de…
2 days ago
BUILD_DEPS.TXT
added rpm dependence to RPMs build
a year ago
CHANGES_McStas
Update release no in one spot..
a month ago
CHANGES_McXtrace
this is not finished
2 months ago

Feature History - and plans for NG (not fully up-to-date)

Beta:

a. First package build of McXtrace Linux and Windows XP
b. 2 Example beamlines
c. Few components

d. Optimized packaging, Linux, Windows7, XP and Mac OSX

1.0:

e. Time-propagation
f. Phase-propagation, wavefront reconstruction experimental
g. Sample models
h. Monochromator crystal (Perfect_crystal)

1.1:

i. Linux, Windows 7, XP, Mac OSX, FreeBSD
j. Optimized grammar
k. Chopper model
l. Faster data file searching
m. Lots more components
n. More Sample models
o. OFF-support - anyshape options enabled
p. Roughness in lenses
q. Shadow interfaces

1.2:

r. improved polarisation handling
s. homogenized source physics
t. SAXS-samples suite
u. web interface
v. Windows 8,8.1

1.4:

v. polycrystal sample
w. New GUI
x. More examples instrument files
y. DEPENDENCY keyword
z. MCPL-interface
aa. Automatic FreeBSD build
ab. SPECTRA source interface
ac. Undulator source

1.5:

ad. Reflectivity library
ae. MAX_Bloch
af. Laue crystal model
ag. MAXIV_DanMAX
ah. MAXIV_FemtoMAX
ai. NIST reflectometer
aj. Reflective grating
ak. Johann Spectrometer
al. Powder Incoherent scattering
am. More accurate Source_lab

Next-generation:

an. GPU suppport (~ 40 x speedup)
ao. Union
ap. Reverse option

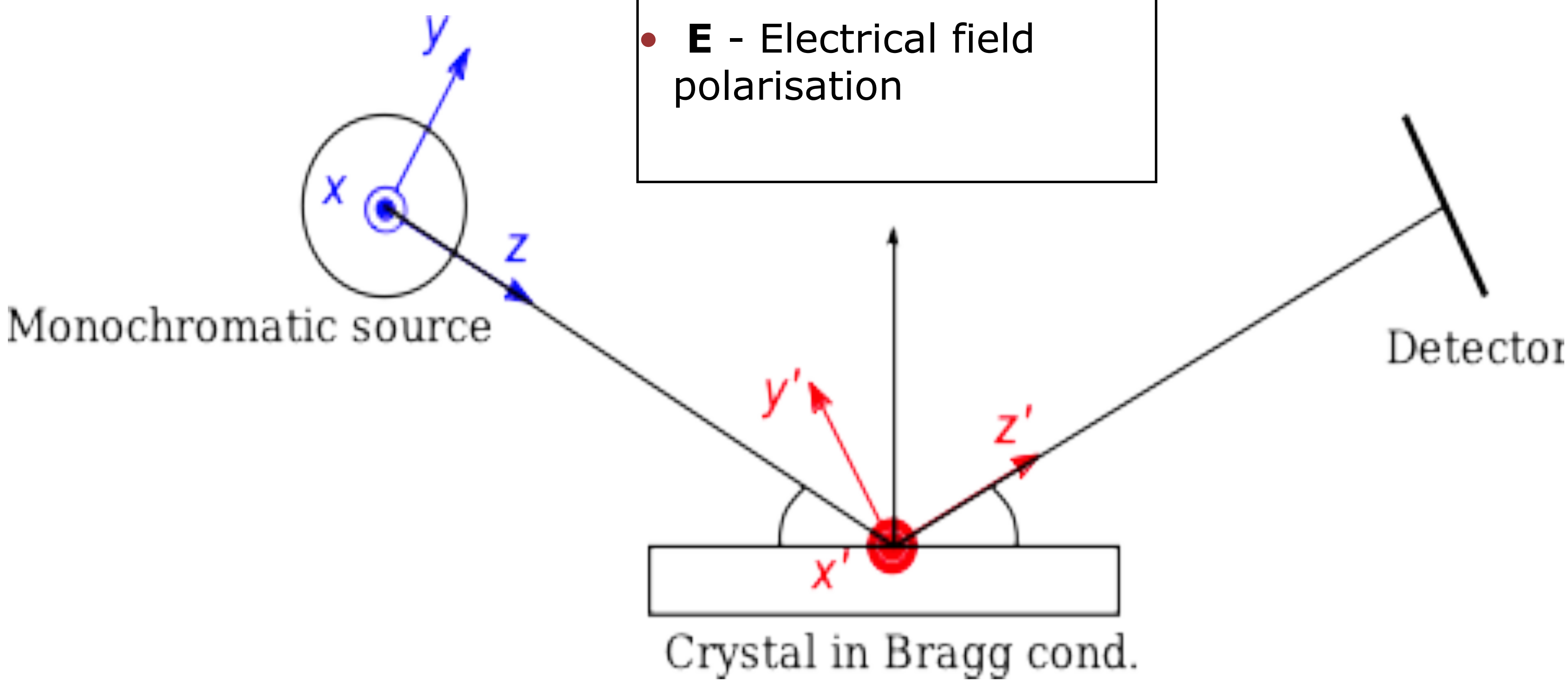
The McXtrace way

1. Describe your beamline in the McXtrace language (In a text file).
2. Automatically convert beamline into ANSI c
3. Compile
4. Run

1. Optimized for your platform
2. Only includes what you use

How?

- Photon ray/package:
- (**r**,**k**,φ,t,p,**E**)
- **r** - spatial coordinates
- **k** - wave vector
- φ - phase
- t - time
- p – photon weight
- **E** - Electrical field polarisation

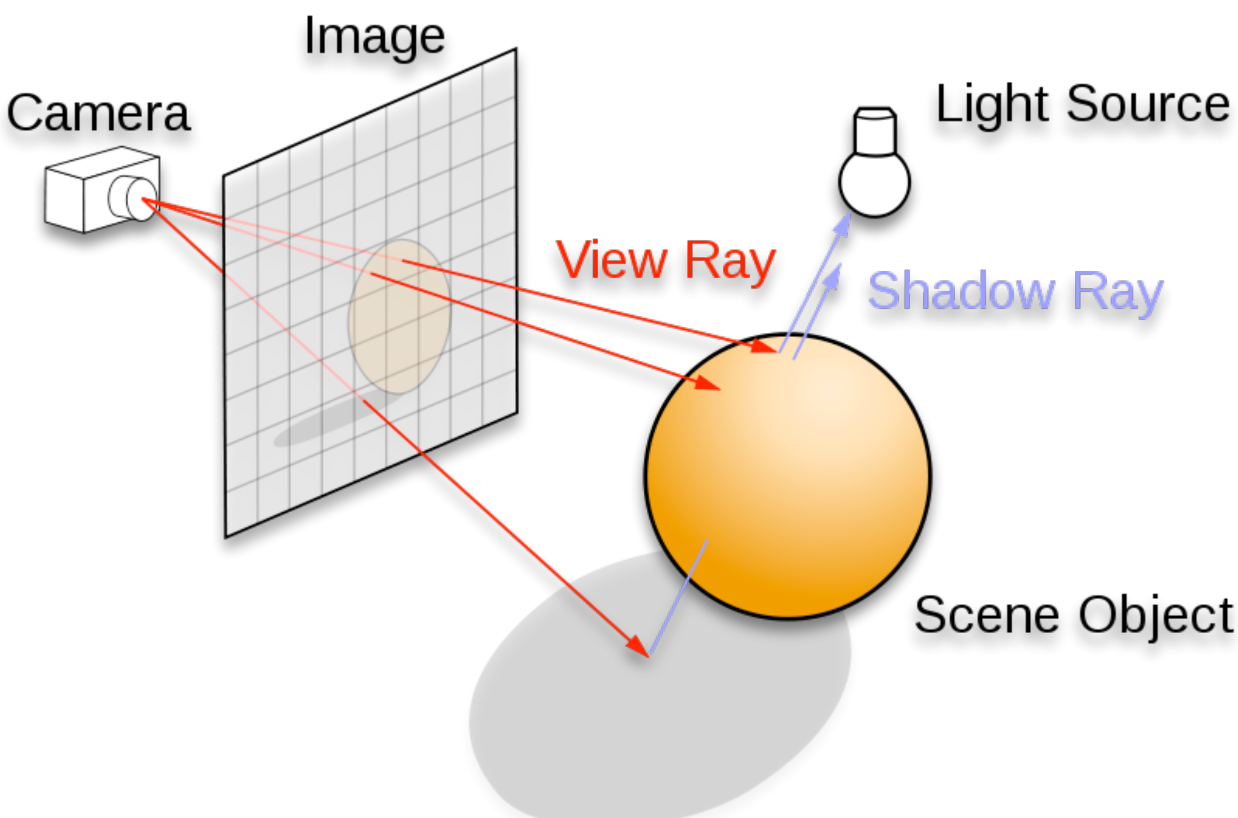


Monte Carlo techniques

- During WW2, “numerical experiments” were applied at Los Alamos for solving mathematical complications of computing fission, criticality, neutronics, hydrodynamics, thermonuclear detonation etc.

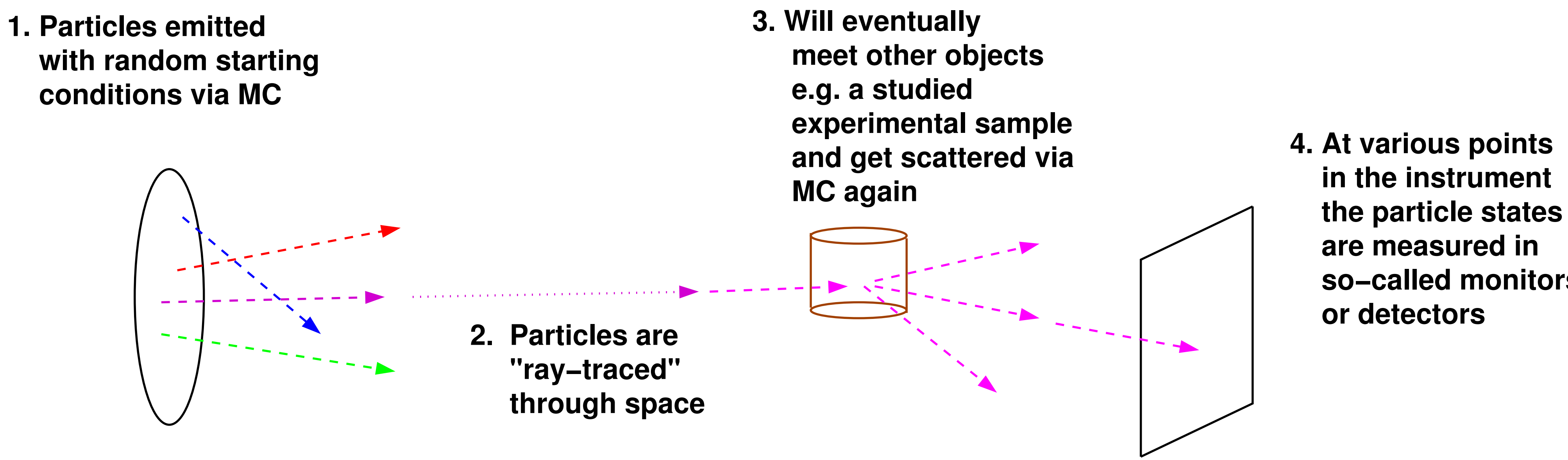
- Notable fathers: John v. Neumann Stanislav Ulam Nicholas Metropolis
- Named “Monte Carlo” after Ulam’s fathers frequent visits to the Monte Carlo casino in Las Vegas
- Initially “implemented” by letting large numbers of women use tabularized random numbers and hand calculators for individual particle calculations
- Later, analogue and digital computing devices were used

Ray-tracing methods



- When particles move in “free space”, we use ray-tracing
  - but in most cases in direction source -> detector

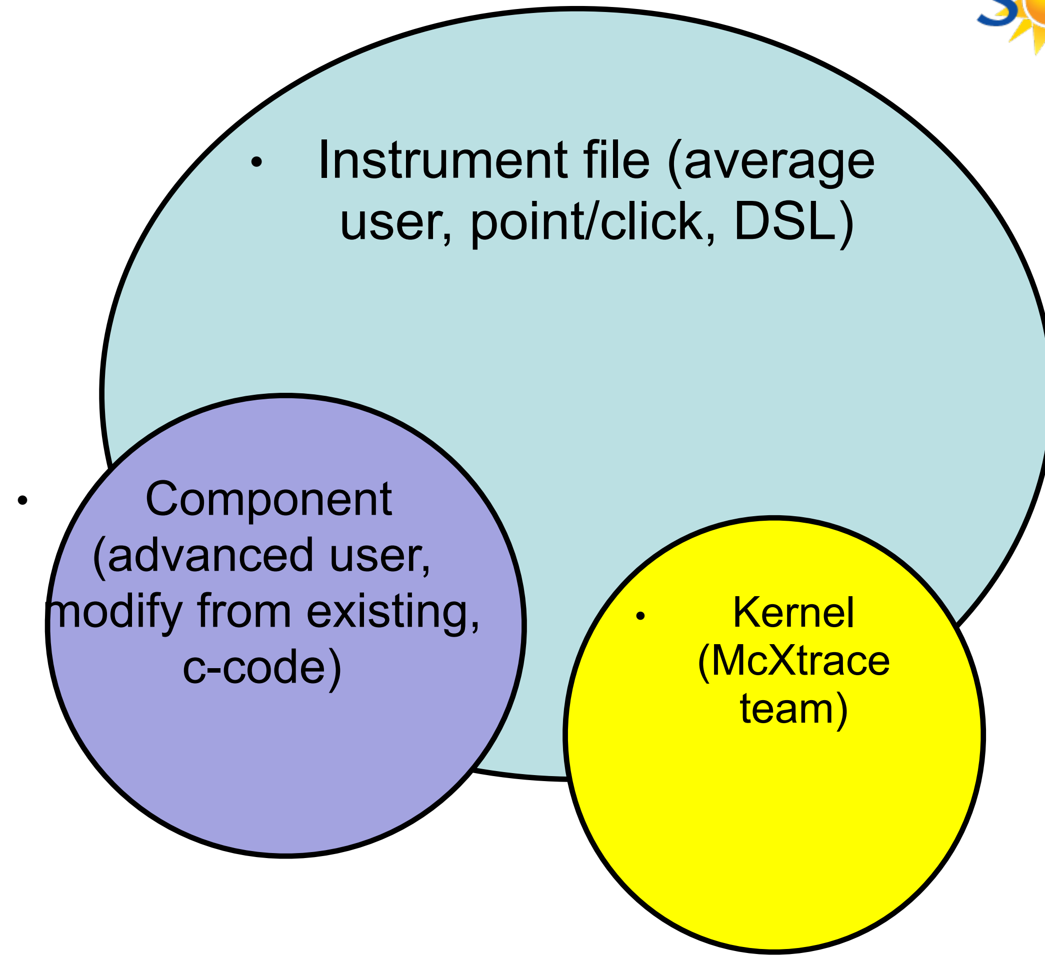
Overall picture: Monte Carlo Ray-Tracing



- Important efficiency mechanisms:
  - “Focusing” - e.g. lab source to first slit (4π vs. limited solid angle only)
  - Rather vs. single particle description, absorption handled though statistics and downscaling the ray weight

Code Levels

- **Instrument file – all users**
  - **existing examples**
  - **user written – GUI assisted**
- Component files – some users
  - Short pieces of code
  - Easy to modify from existing
- Kernel code – McXtrace developers
  - Propagation routines
  - Intersections
- Generated ISO-C code – “no” users
  - Assembled by code generation
  - Very low overhead of unneeded code
  - Includes runtime libs that comps rely on (propagation etc.)



Instrument file

JJ_SAXS.instr + (~/Repositories/McCode/mcxtrace-comps/examples) - GVIM

File Edit Tools Syntax Buffers Window Help

```
/*******************************************************************************
*         McXtrace instrument definition URL=http://www.mcxtrace.org
*
* Instrument: SAXS_saxslab (rename also the example and DEFINE lines below)
*
* %Identification
* Written by: Erik Knudsen (erkn@risoe.dtu.dk)
* Date: September 24th, 2009
* Origin: Ris&oslash; DTU, (Finnair flight AY67 to Hong Kong)
* Release: McXtrace
* Version: 0.1_alpha
* %INSTRUMENT_SITE: SAXSLAB
*
* Crude model of a laboratory SAXS-instrument mimicking the type sold by SAXSlab/JJ-Xray Systems.
*
* %Description
*
*
* %Parameters
* pin2_pos: [m] distance between 1st and 2nd pinhole in beam tube
* pin3_pos: [m] distance between 2nd and 3rd pinhole in beam tube
* optic_L:  [m] length of the focusing optic
* sample_pos:[m] distance from 3rd pinhole to sample
* detector_pos: [m] distance from 3rd pinhole to detector
*
* %Example: pin2_pos=0.2 pin3_pos=0.4 optic_L=0.1 sample_pos=0.2 detector_pos=2 Detector: psd1_I=7.8629e-05
*
* %Link
* A reference/HTML link for more information
*
* %End
*******************************************************************************/

DEFINE INSTRUMENT SAXS_saxlab(pin2_pos=0.2, pin3_pos=0.4, optic_L=0.1, sample_pos=0.2, detector_pos=2)

DECLARE
%{
%}

INITIALIZE
%{
%}

TRACE

COMPONENT Origin = Progress_bar()
  AT (0,0,0) ABSOLUTE

COMPONENT apparent_source=Source_flat(
    xwidth=8e-3,yheight=.04e-3,dist=0.1,focus_yh=0.001,focus_xw=0.01,lambda0=1.54,dlambda=0.1
)
AT (0,0,0) RELATIVE PREVIOUS

COMPONENT psd00=PSD_monitor(
    filename="psd00.dat",xwidth=0.2,yheight=0.2,restore_xray=1
)
  AT(0,0,0.0999) RELATIVE PREVIOUS

  COMPONENT optic_arm=Arm()
AT(0,0,0.1) RELATIVE apparent_source
ROTATED(0,0,-90) RELATIVE apparent_source

COMPONENT optic=Mirror_curved(
    radius=20,length=0.2)
AT(0,0,0) RELATIVE optic_arm
ROTATED(0,2.75,0) RELATIVE optic_arm
EXTEND
%{
  if (!SCATTERED) ABSORB;
%}
```

Repositories/McCode/mcxtrace-comps/examples/JJ_SAXS.instr [+]   55,1   Top

```
COMPONENT beam_tube_axis=Arm()
AT(0,0,0.1) RELATIVE apparent_source
ROTATED (5.5,0,0) RELATIVE Origin

COMPONENT pinhole1=Slit(xwidth=0.1,yheight=0.1)
  AT(0,0,0.15) RELATIVE beam_tube_axis

COMPONENT psd01=PSD_monitor(
    filename="psd01.dat",xwidth=0.2,yheight=0.2,restore_xray=1
)
  AT(0,0,1e-3) RELATIVE PREVIOUS

COMPONENT pinhole2=Slit(xwidth=0.1,yheight=0.1)
  AT(0,0,pin2_pos) RELATIVE beam_tube_axis

/*COMPONENT psd02=PSD_monitor(
    filename="psd02.dat",xwidth=0.2,yheight=0.2
)
AT (0,0,1e-3) RELATIVE PREVIOUS
*/
COMPONENT pinhole3=Slit(xwidth=0.1,yheight=0.1)
  AT(0,0,pin3_pos) RELATIVE beam_tube_axis
COMPONENT psd03=PSD_monitor(
    filename="psd02.dat",xwidth=0.2,yheight=0.2
)
AT (0,0,1e-3) RELATIVE PREVIOUS

COMPONENT psd1=PSD_monitor(
    filename="psd1.dat",xwidth=0.4,yheight=0.4
)
AT (0,0,2) RELATIVE psd03

/* This section is executed when the simulation ends (C code). Other    */
/* optional sections are : SAVE                                         */
FINALLY
%{
%}
/* The END token marks the instrument definition end */
END
```

Repositories/McCode/mcxtrace-comps/examples/JJ_SAXS.instr [+]   141,1   Bot

-- INSERT --

Component file

PSD_monitor.comp (~/Repositories/McCode/mcxtrace-comps/monitors) - GVIM10

File Edit Tools Syntax Buffers Window Help

```
DEFINE COMPONENT PSD_monitor
DEFINITION PARAMETERS (nx=90, ny=90,nr=0, string filename=0, restore_xray=0)
  SETTING PARAMETERS (xmin=-0.05, xmax=0.05, ymin=-0.05, ymax=0.05, xwidth=0, yheight=0, radius=0)
OUTPUT PARAMETERS (PSD_N, PSD_p, PSD_p2)
/* X-ray parameters: (x,y,z,kx,ky,kz,phi,t,Ex,Ey,Ez,p) */

DECLARE
  %{
    double **PSD_N;
    double **PSD_p;
    double **PSD_p2;
  %}
INITIALIZE
  %{
    int i,j;
    double *p1,*p2,*p3;

    if (xwidth  > 0) { xmax = xwidth/2;  xmin = -xmax; }
    if (yheight > 0) { ymax = yheight/2; ymin = -ymax; }

    if ( ((xmin >= xmax) || (ymin >= ymax)) && !radius ) {
            printf("PSD_monitor: %s: Null detection area !\n"
                "ERROR        (xwidth,yheight,xmin,xmax,ymin,ymax,radius). Exiting",
           NAME_CURRENT_COMP);
      exit(0);
    }
    if(!radius){
      p1=calloc(nx*ny,sizeof(double));
      p2=calloc(nx*ny,sizeof(double));
      p3=calloc(nx*ny,sizeof(double));

      PSD_N=calloc(nx,sizeof(double *));
      PSD_p=calloc(nx,sizeof(double *));
      PSD_p2=calloc(nx,sizeof(double *));

      for (i=0; i<nx; i++){
        PSD_N[i]=&(p1[i*ny]);//calloc(ny,sizeof(double));
        PSD_p[i]=&(p2[i*ny]);//calloc(ny,sizeof(double));
        PSD_p2[i]=&(p3[i*ny]);//calloc(ny,sizeof(double));
      }
    }else{
      PSD_N=calloc(1,sizeof(double *));
      PSD_p=calloc(1,sizeof(double *));
      PSD_p2=calloc(1,sizeof(double *));
      *PSD_N=calloc(nr,sizeof(double));
      *PSD_p=calloc(nr,sizeof(double));
      *PSD_p2=calloc(nr,sizeof(double));
    }

  %}
TRACE
  %{
    int i,j;

    PROP_Z0;
    if (!radius){
      if (x>xmin && x<xmax && y>ymin && y<ymax)
      {
        i = floor((x - xmin)*nx/(xmax - xmin));
        j = floor((y - ymin)*ny/(ymax - ymin));
        PSD_N[i][j]++;
        PSD_p[i][j] += p;
        PSD_p2[i][j] += p*p;
        SCATTER;
      }
    }else{
      double r=sqrt(x*x+y*y);
      if (r<radius){
        i = floor(r*nr/radius);
        PSD_N[0][i]++;
        PSD_p[0][i] += p;
../McCode/mcxtrace-comps/monitors/PSD_monitor.comp          75,61          52%
```

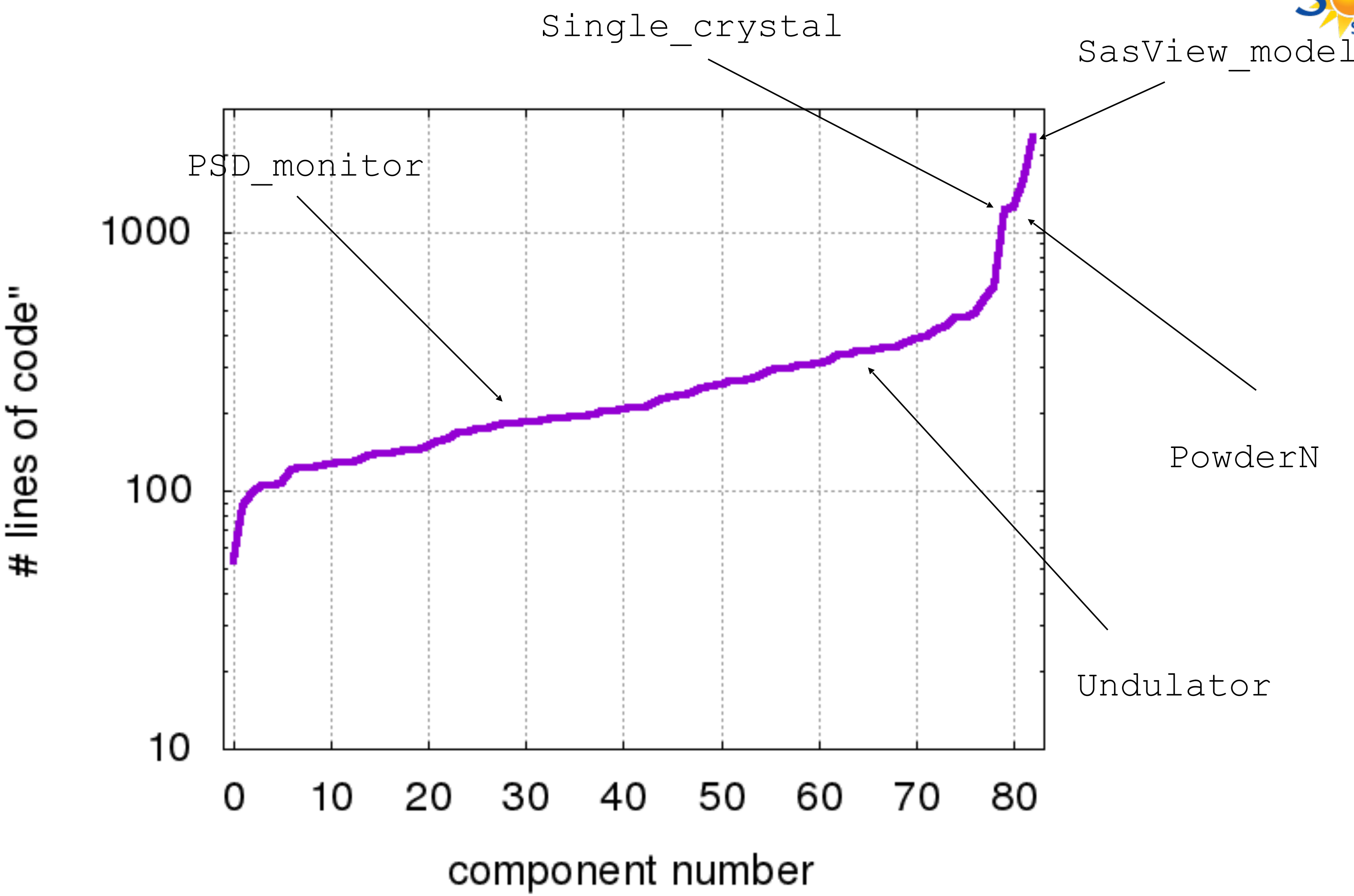
```
        j = floor((y - ymin)*ny/(ymax - ymin));
        PSD_N[i][j]++;
        PSD_p[i][j] += p;
        PSD_p2[i][j] += p*p;
        SCATTER;
      }
    }else{
      double r=sqrt(x*x+y*y);
      if (r<radius){
        i = floor(r*nr/radius);
        PSD_N[0][i]++;
        PSD_p[0][i] += p;
        PSD_p2[0][i] += p*p;
        SCATTER;
      }
    }
    if (restore_xray) {
      RESTORE_XRAY(INDEX_CURRENT_COMP, x, y, z, kx, ky, kz, phi, t, Ex, Ey, Ez, p);
    }
  %}
SAVE
  %{
    if(!radius){
      DETECTOR_OUT_2D(
          "PSD monitor",
          "X position [m]",
          "Y position [m]",
          xmin, xmax, ymin, ymax,
          nx, ny,
          *PSD_N,*PSD_p,*PSD_p2,
          filename);
    }else{
      DETECTOR_OUT_1D(
          "PSD_monitor","Radial Position[m]", "Intensity", "R",
          0,radius,nr,&PSD_N[0][0],&PSD_p[0][0],&PSD_p2[0][0],filename);
    }

  %}

FINALLY
%{
   free(PSD_N[0]);
   free(PSD_N);
   free(PSD_p[0]);
   free(PSD_p);
   free(PSD_p2[0]);
   free(PSD_p2);
%}
MCDISPLAY
%{
  magnify("xy");
  multiline(5, (double)xmin, (double)ymin, 0.0,
               (double)xmax, (double)ymin, 0.0,
               (double)xmax, (double)ymax, 0.0,
               (double)xmin, (double)ymax, 0.0,
               (double)xmin, (double)ymin, 0.0);


END
../McCode/mcxtrace-comps/monitors/PSD_monitor.comp          131,5          Bot
```

Component size



Why bother?

1) Why simulate when you can just do the experiment?

2) Why McXtrace and not sim. package X?

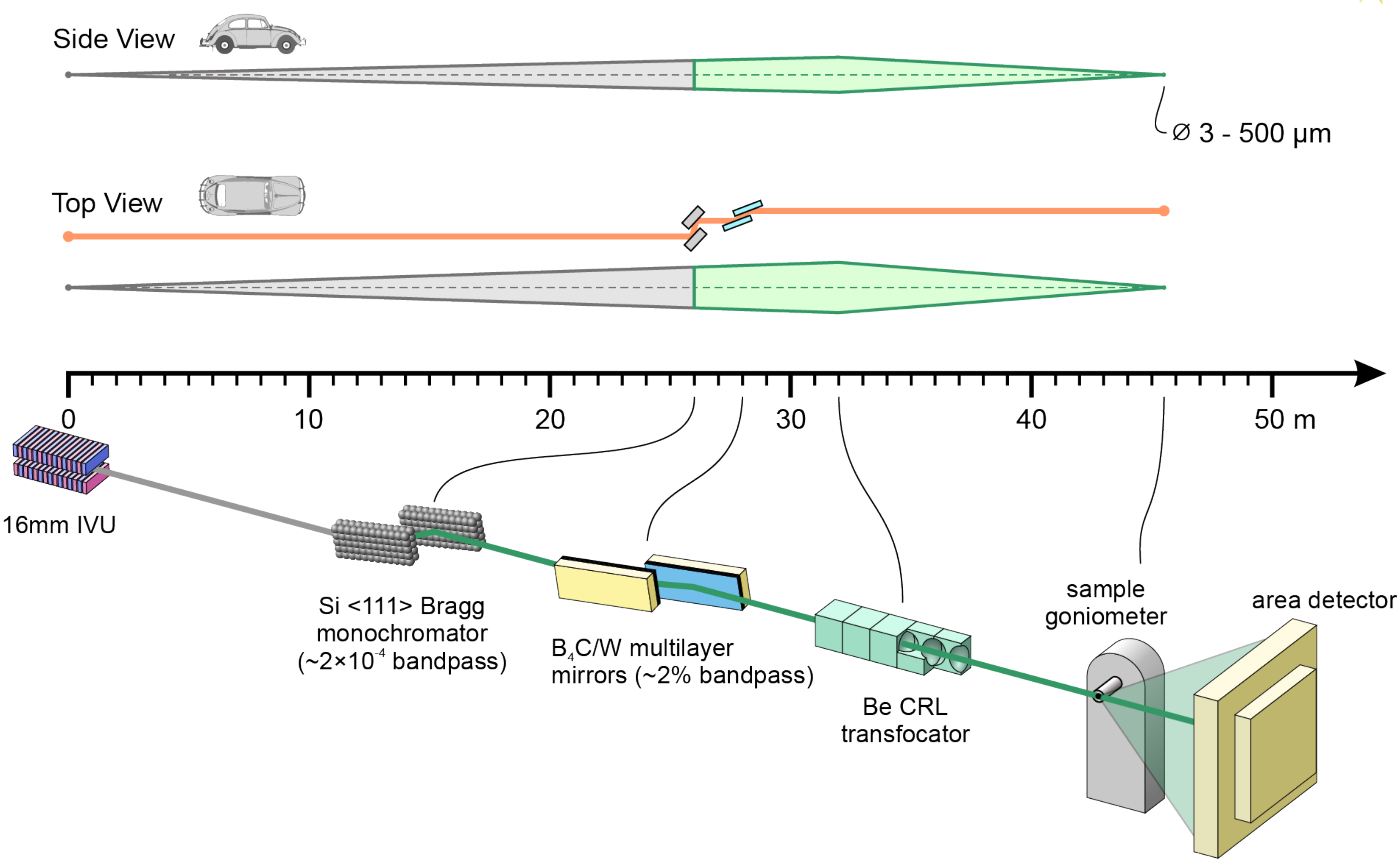
Why simulate a full beamline?

- Users don't have the luxury you have!
  - Prospective users may try out experiments.
  - Teach users how to do experiments.

Why McXtrace

- Community driven code
- Fairly easy to extend – write a new component and it automatically fits into the framework.
- Portable (need a c-compiler).
- Several included standard sample models.
- Once written, a simulation/beamline is a real program in itself.
- ...but...
- Use whatever you prefer! XRT, SHADOW, SRW, Ray,...<insert package here>...

DanMAX



Side View
Ø 3 - 500 µm
Top View
0
10
20
30
40
50 m
16mm IVU
Si <111> Bragg monochromator (~2×10^-4 bandpass)
B_4C/W multilayer mirrors (~2% bandpass)
Be CRL transfocator
sample goniometer
area detector

DanMAX web interface

Instrument - Chromium

Instrument | Simulation Data

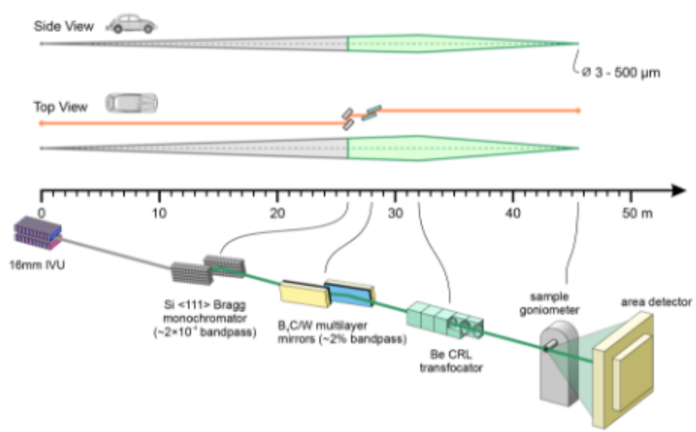
https://sim.e-neutrons.org/instrument/xrd/DanMAX

Apps | Debian.org | Latest News | Help

e-neutrons McStas

Logged in as ebknudsen (see recent simruns) Logout

DanMAX (click for documentation)



Parameters for DanMAX

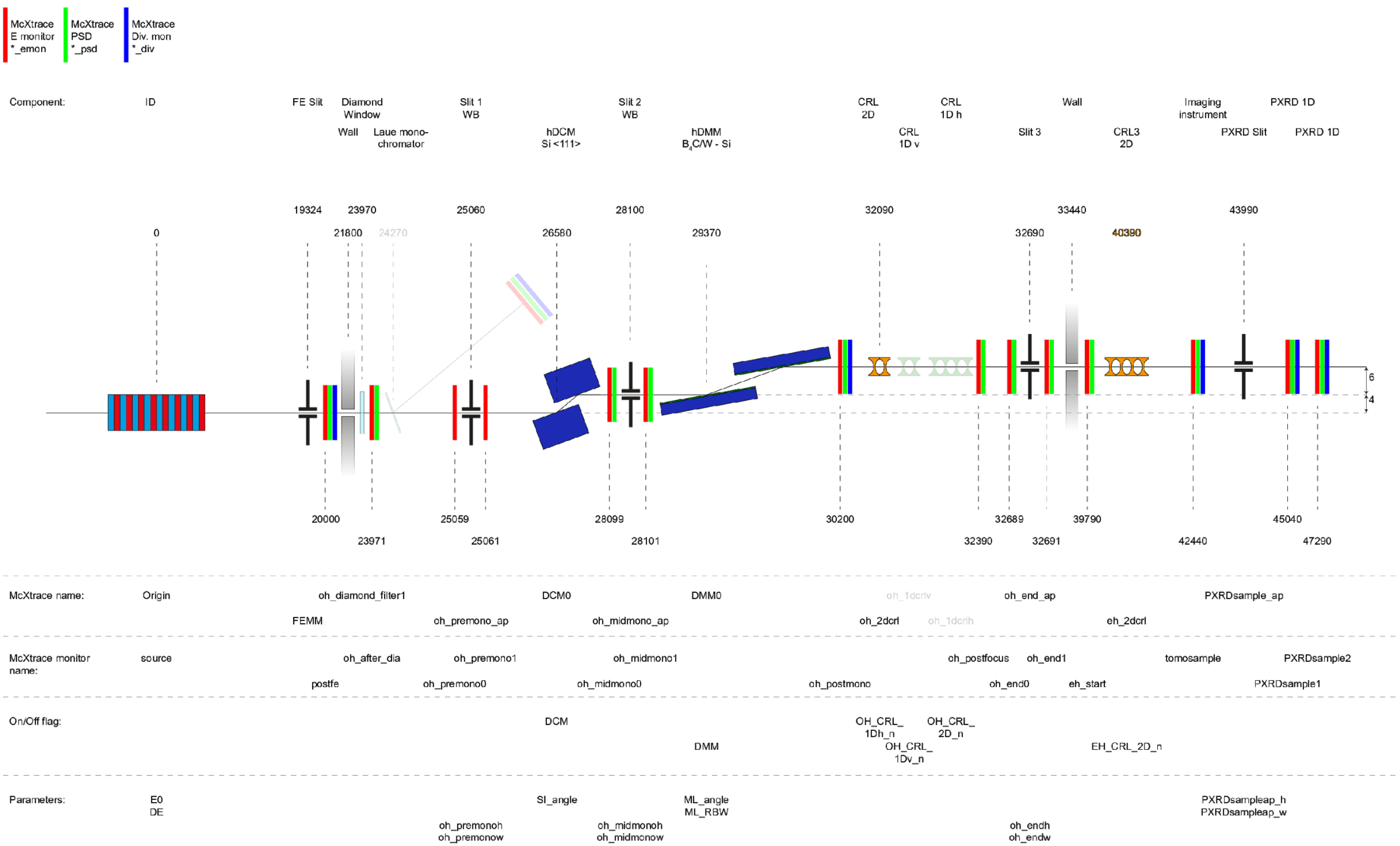
spectra_sfiles [] :	/usr/local/mcxtrace/1.2/data/(	Filename stem (comp will add -x.dta) for source files containing output from spectra calc. (/usr/local/mcxtrace/1.2/data/dm_2d_15keVpm0p5)
SRC [] :	2	Choose which source model to use (2)
E0 [keV] :	15	The central energy to sample from source model. (15)
DE [keV] :	0.5	Spectral width (std. dev. if gaussian source) to sample from source model. (0.5)
DCM_e0 [keV] :	15	The energy to tune the Si monochromator to. May be different from E0. If 0 the monochromator is controlled by DCM_angle. (15)
DMM_e0 [keV] :	15	The energy to tune the Multilayer monochromator to. May be different from E0. If 0 the monochromator is controlled by DMM_angle. (15)
oh_premonoh [m] :	0.005	Vertical (height) opening of aperture in front of the monochromator. (0.005)
oh_premonow [m] :	0.005	Horizontal (width) opening of aperture in front of the monochromator. (0.005)
oh_midmonoh [m] :	0.005	Vertical (height) opening of aperture in the middle of the monochromator (between DCM and DMM). (0.005)
oh_midmonow [m] :	0.005	Horizontal (width) opening of aperture in the middle of the monochromatori (between DCM and DMM). (0.005)
oh_endh [m] :	0.005	Vertical (height) opening of aperture at the end of the optical hutch. (0.005)
oh_endw [m] :	0.005	Horizontal (width) opening of aperture at the end of the optical hutch. (0.005)
PXRDsampleap_h [m] :	0.002	Vertical (height) opening of the guard aperture in front of the PXRD sample. (0.002)

DanMAX simulation effort

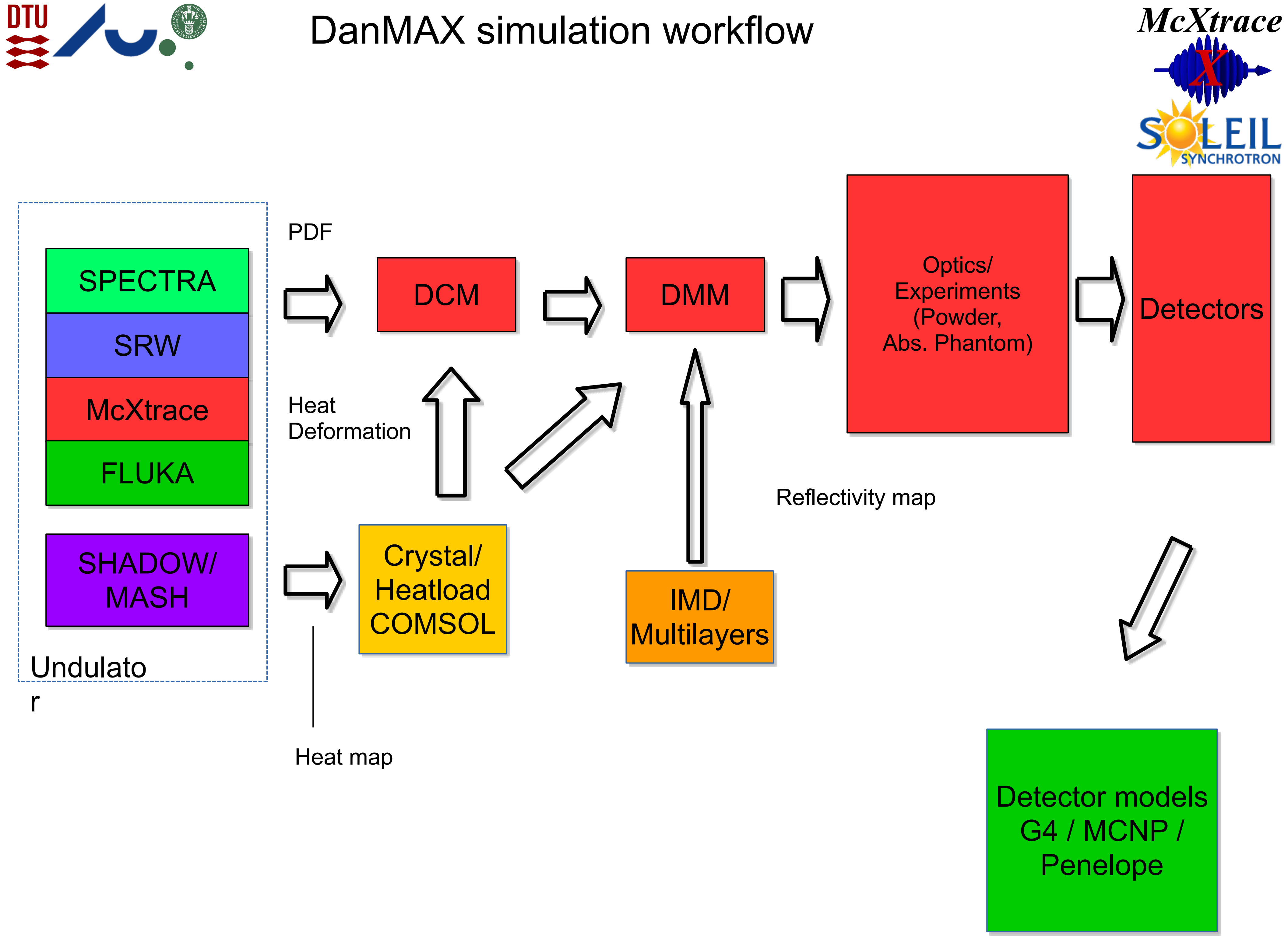
McXtrace main simulation vehicle with connections to:

- SPECTRA / SRW for source (undulator) calculation benchmarking
- COMSOL + SHADOW + MASH – heatload calculations
- IMD – multilayer reflectivity calculations

DanMAX - McXtrace model - cheat sheet

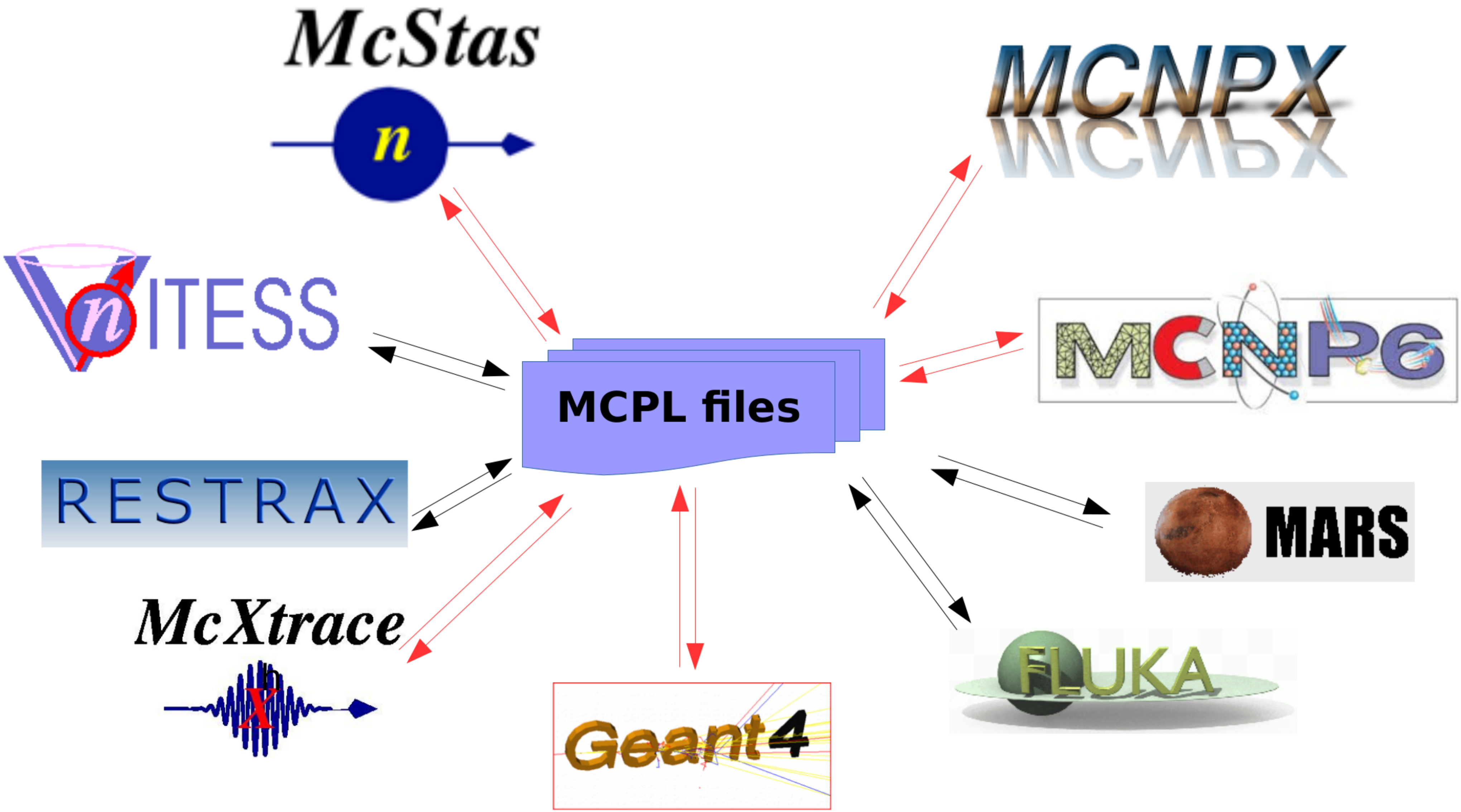


DanMAX simulation workflow



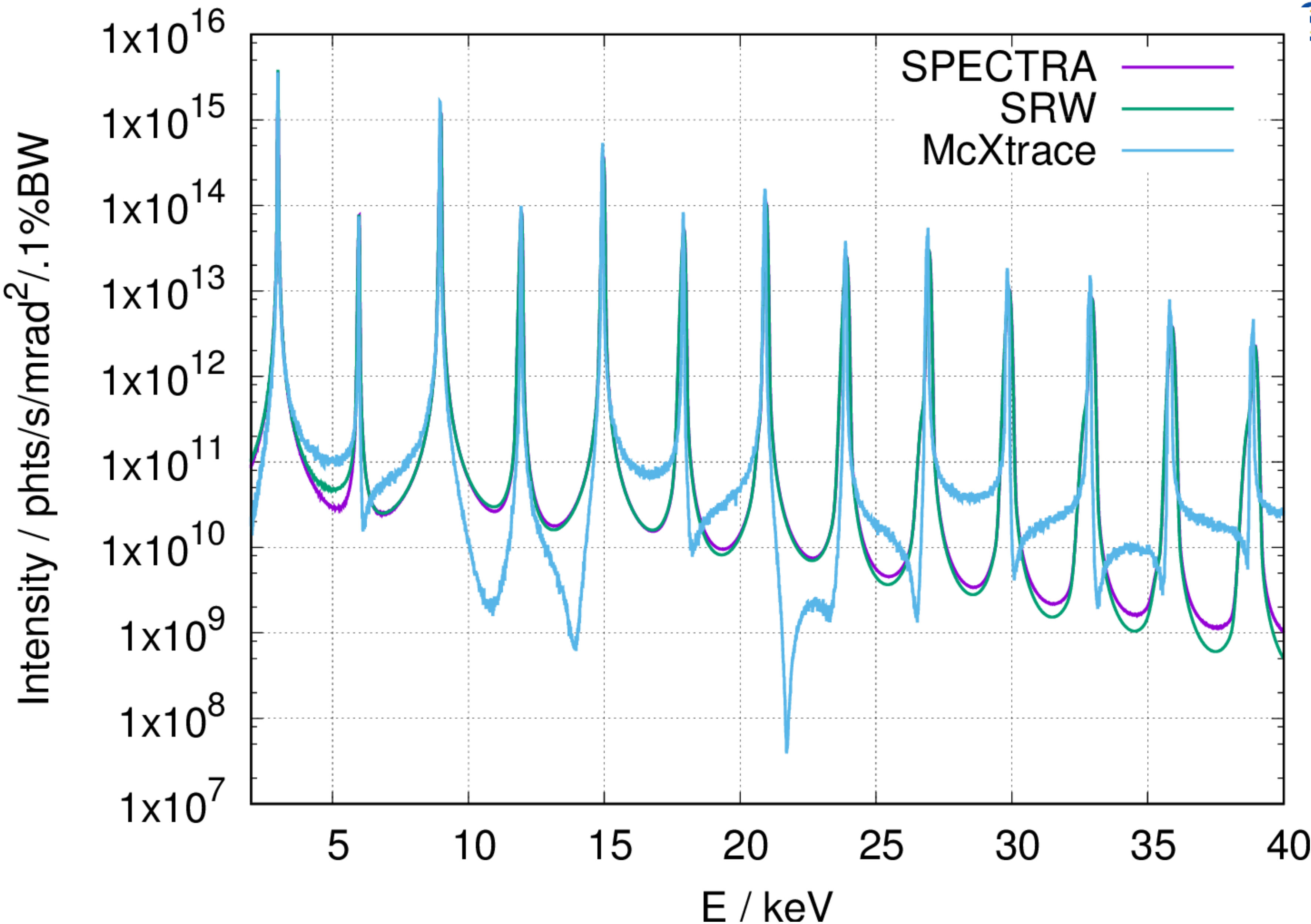
MCPL

Monte Carlo Particle List-format



Courtesy T. Kittelmann, ESS

New Source: Undulator

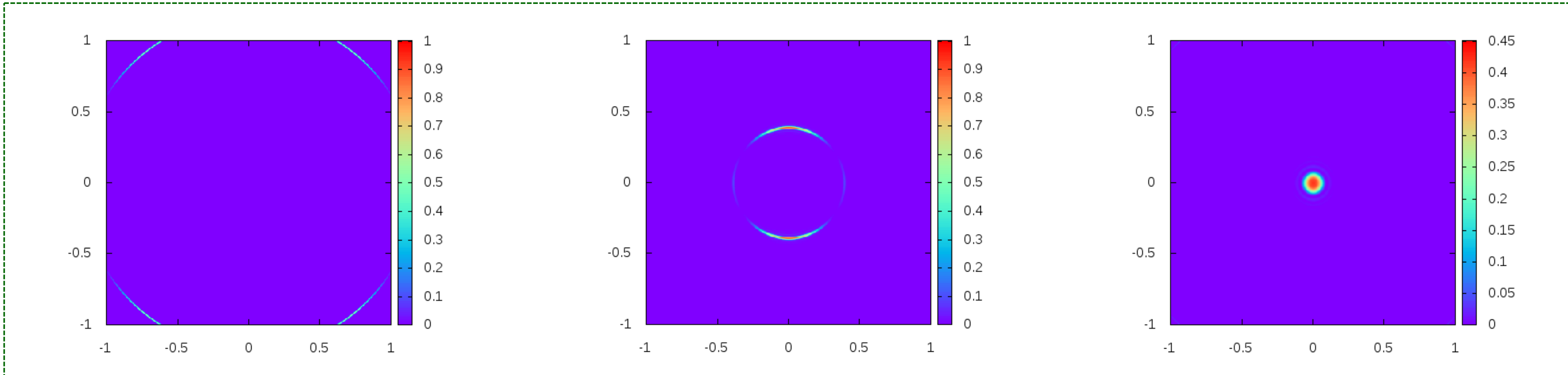


Undulator: Spatial distr.

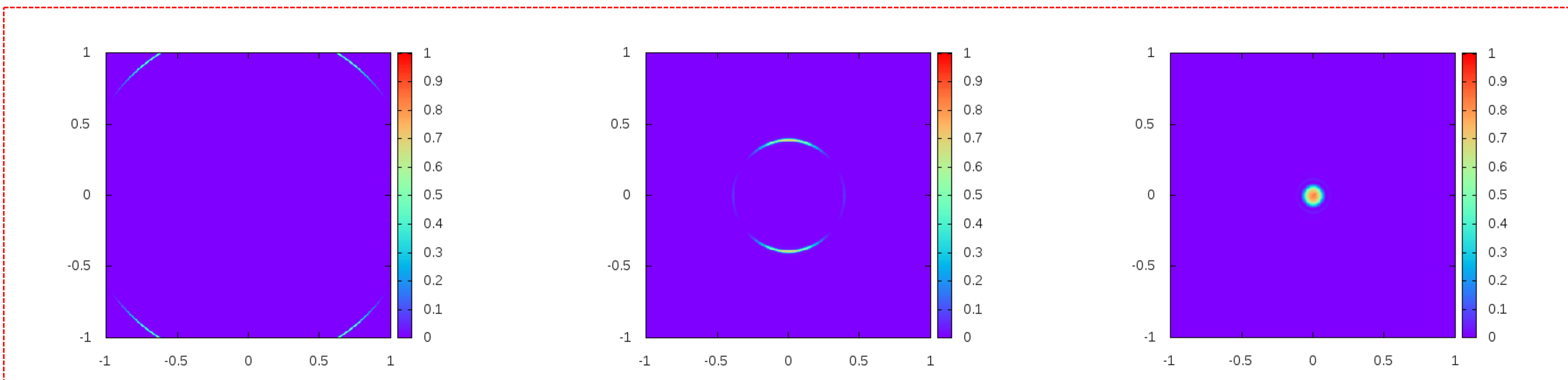
Tuned to have 11th Harmonic at 33 keV

25.3 keV | 26.8 keV | 33 keV

SPECTRA



McXtrace



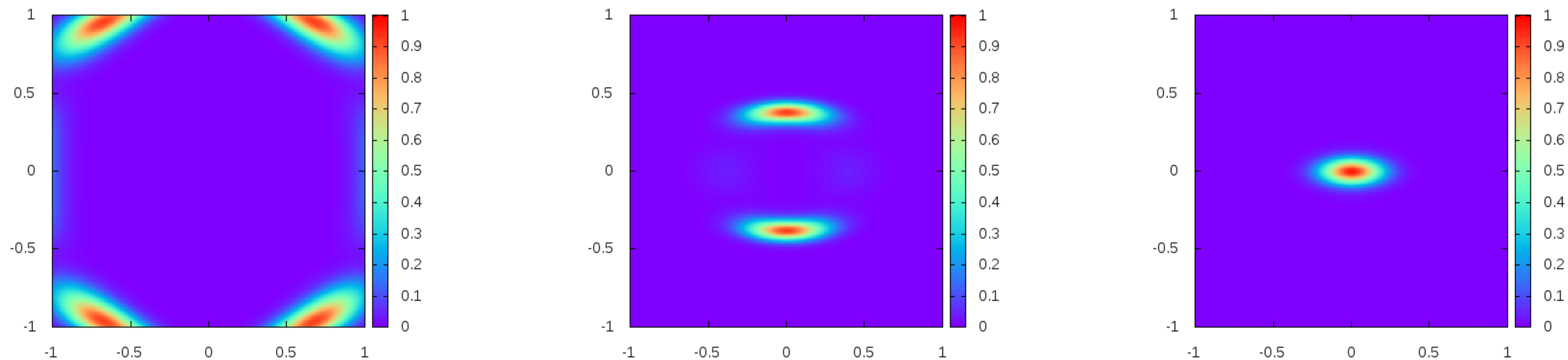
Undulator: Spatial distr.

Tuned to have 11th Harmonic at 33 keV

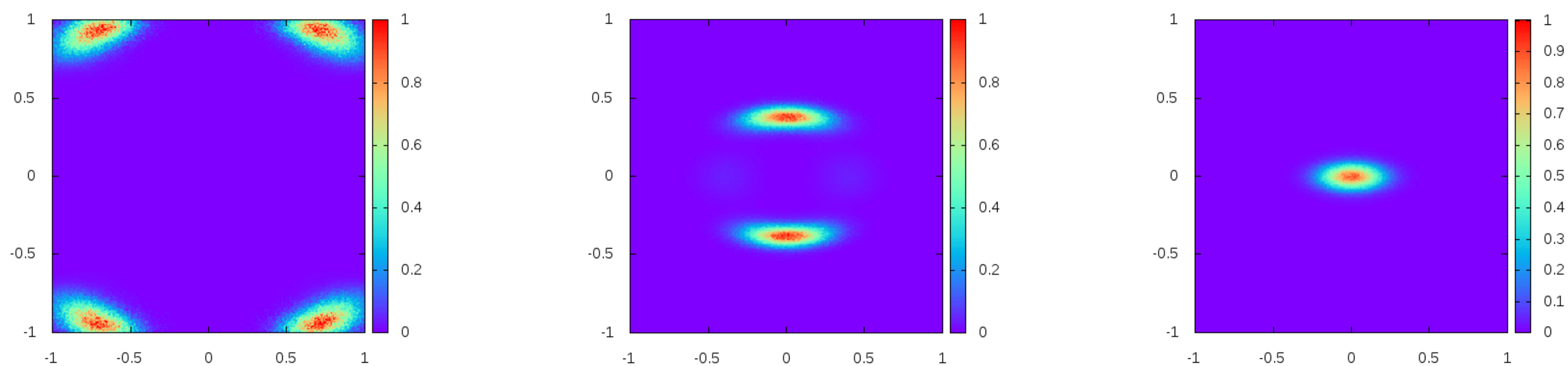
With emittance

25.3 keV | 26.8 keV | 33 keV

SPECTRA



McXtrace

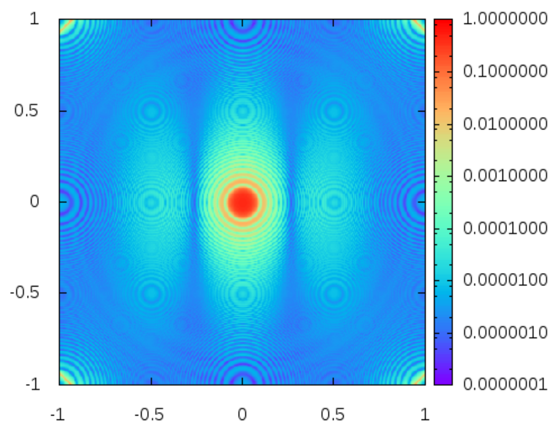


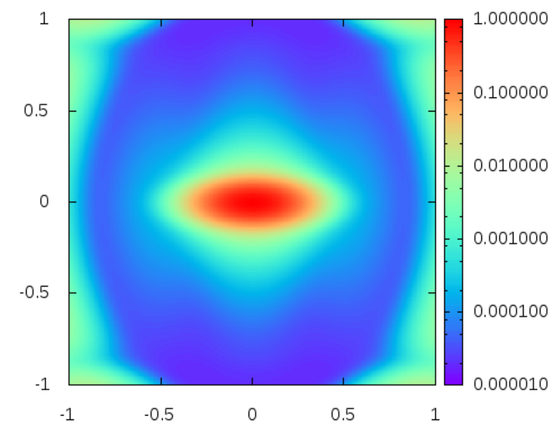
Undulator: Spatial distr.

Log-plots. 33 keV

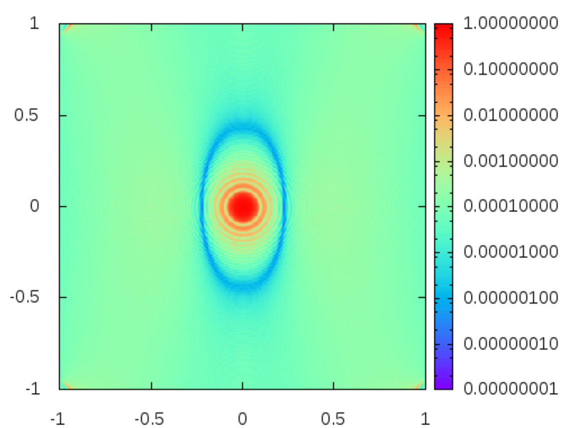
No emittance | With emittance

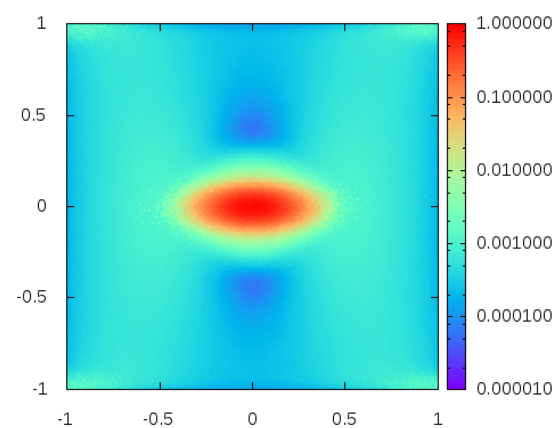
SPECTRA



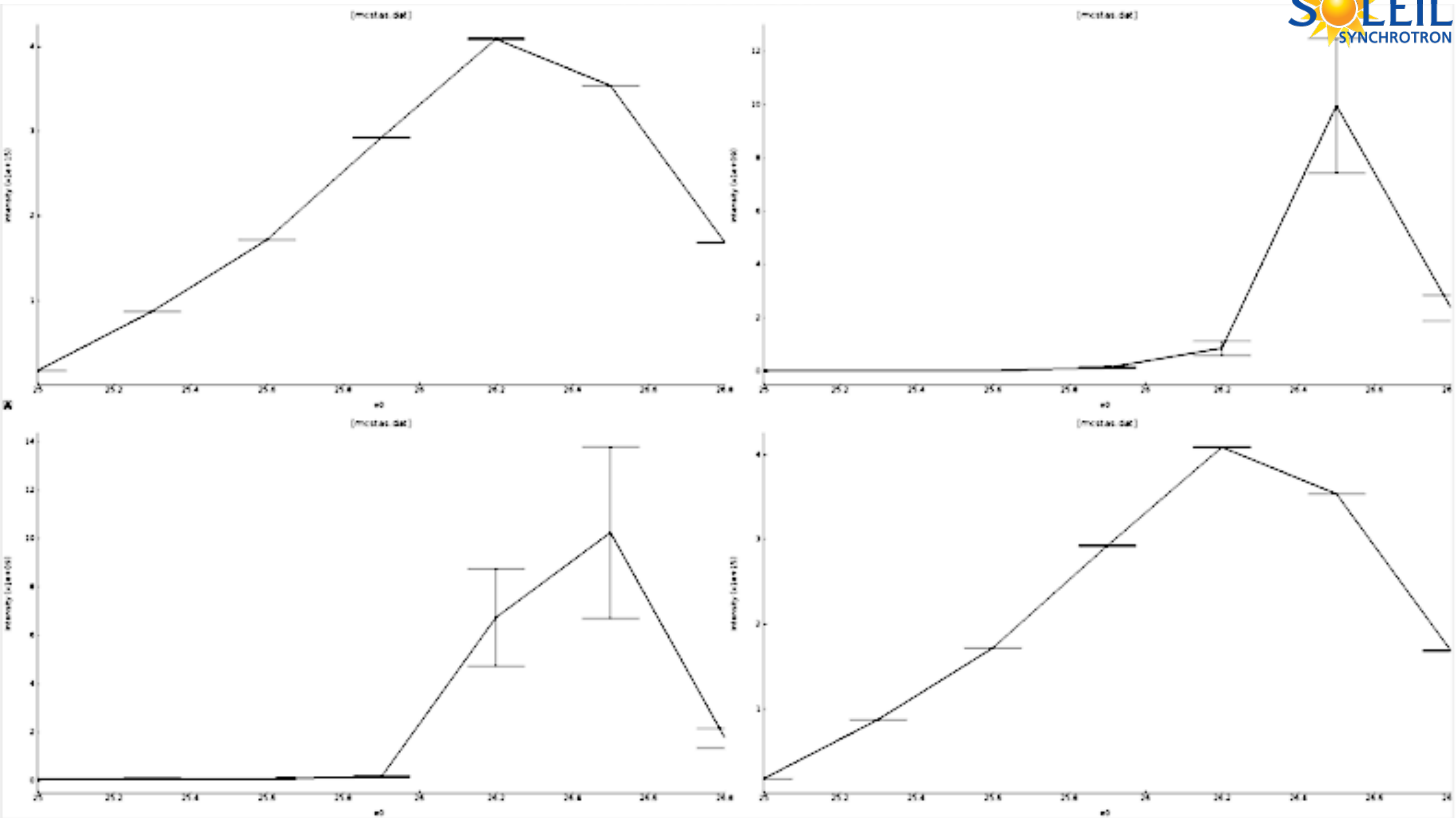


McXtrace

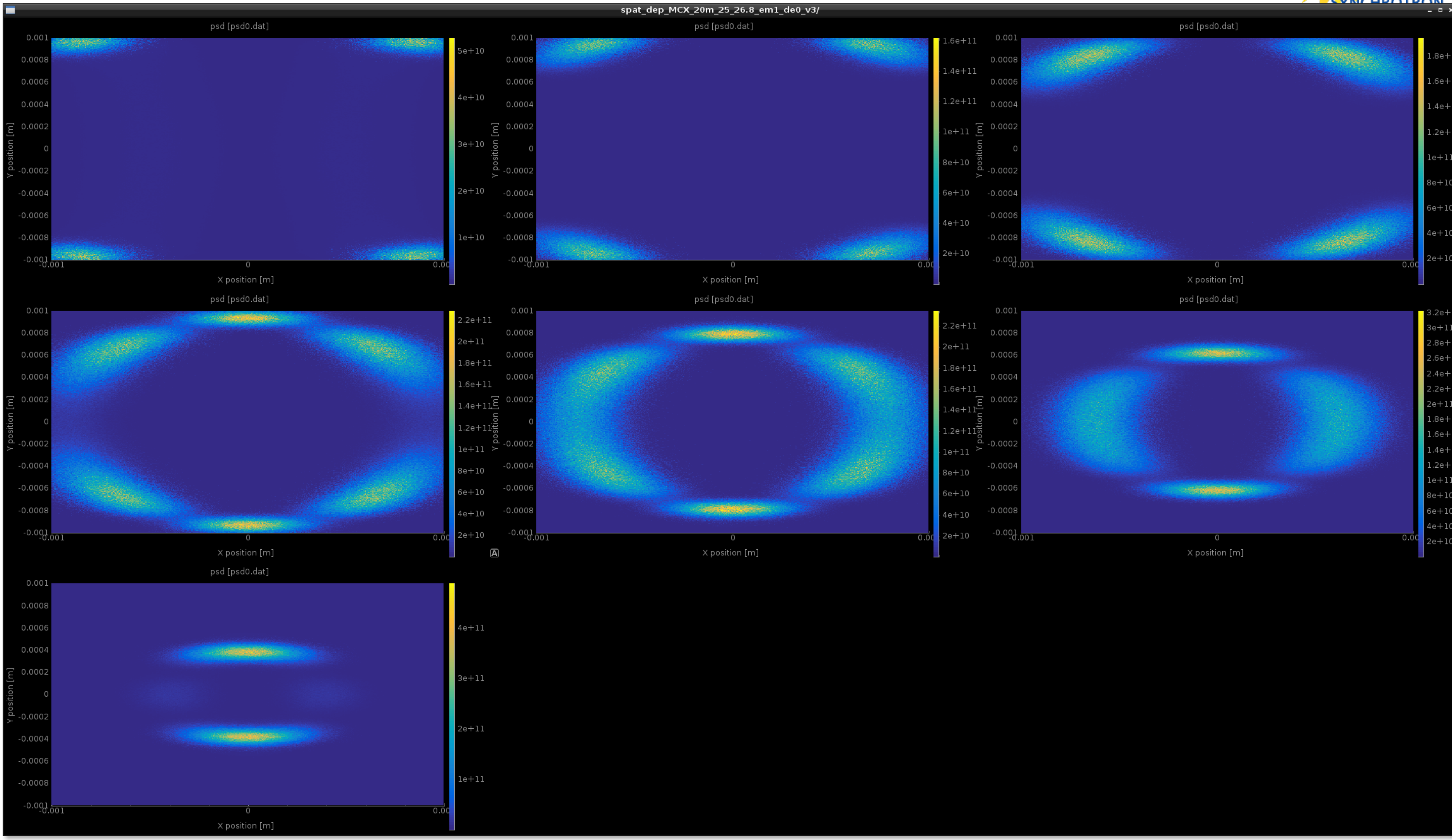




Plot scan results



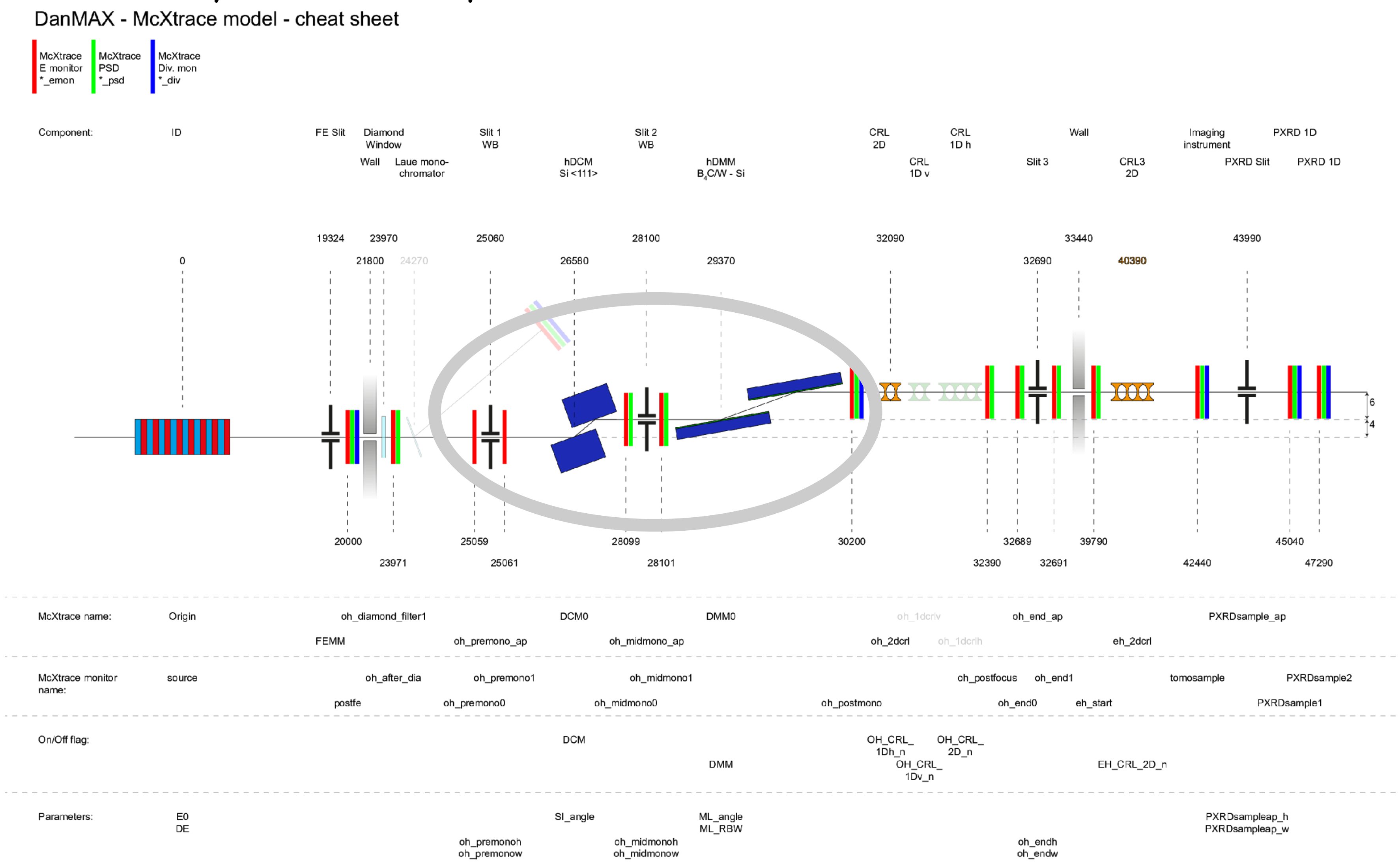
Plot scan results



DanMAX simulation effort

McXtrace main simulation vehicle with connections to:

- SPECTRA / SRW for source (undulator) calculation benchmarking
- COMSOL + SHADOW + MASH – heatload calculations
- IMD – multilayer reflectivity calculations

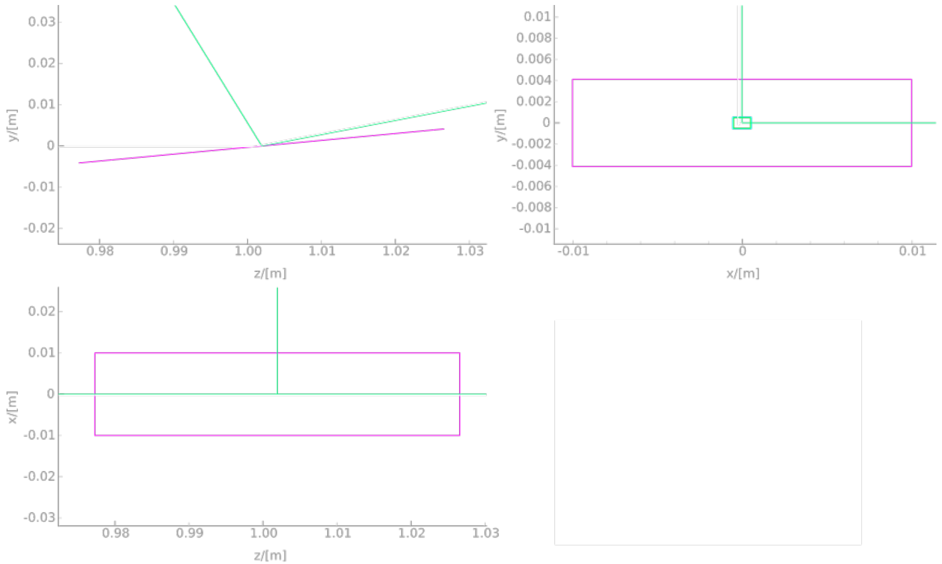


Monochromator Crystals

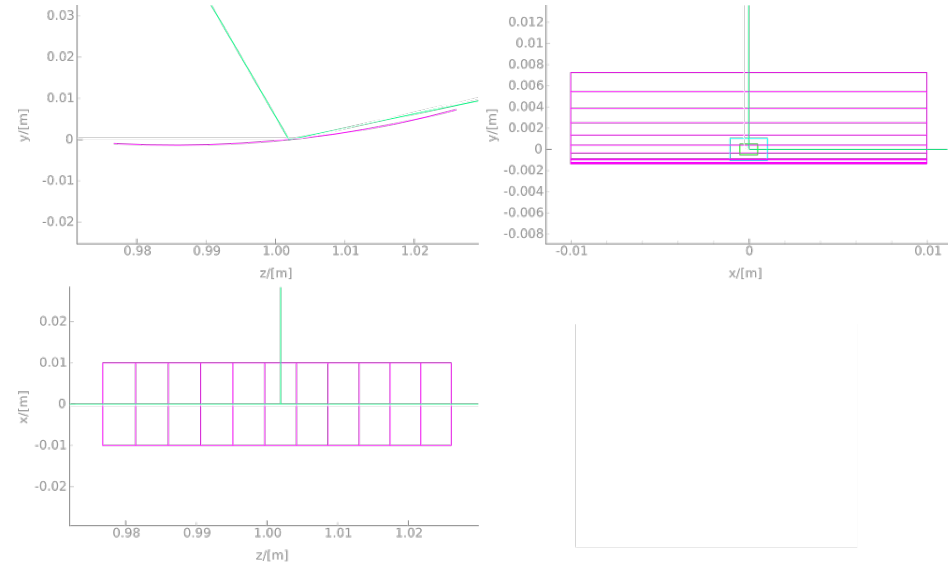
McXtrace

SOLEIL SYNCHROTRON

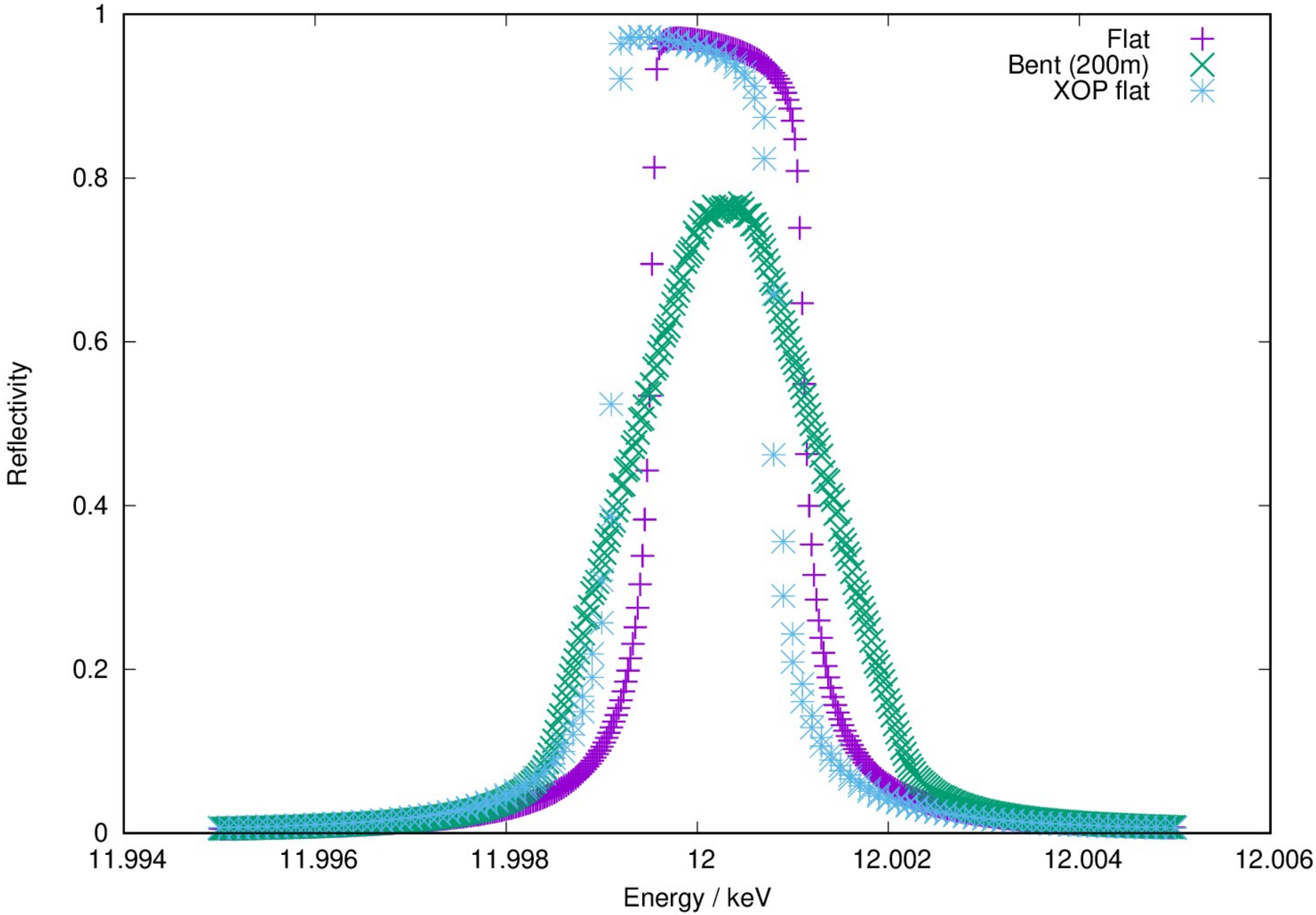
Flat



Bent (r=200m)



Monochromator Crystals

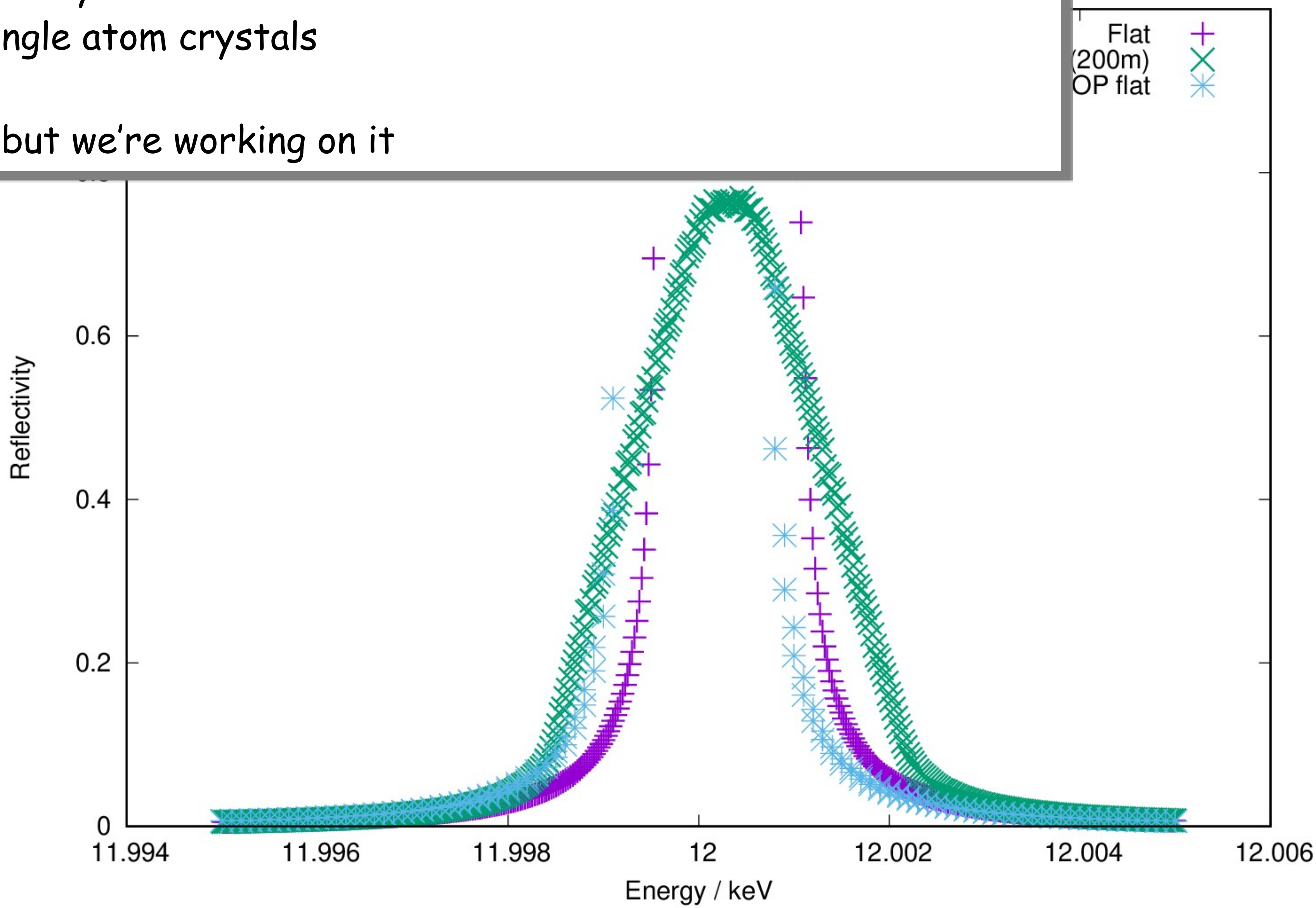


Monochromator Crystals

Currently limited to

- Single atom crystals

… but we're working on it

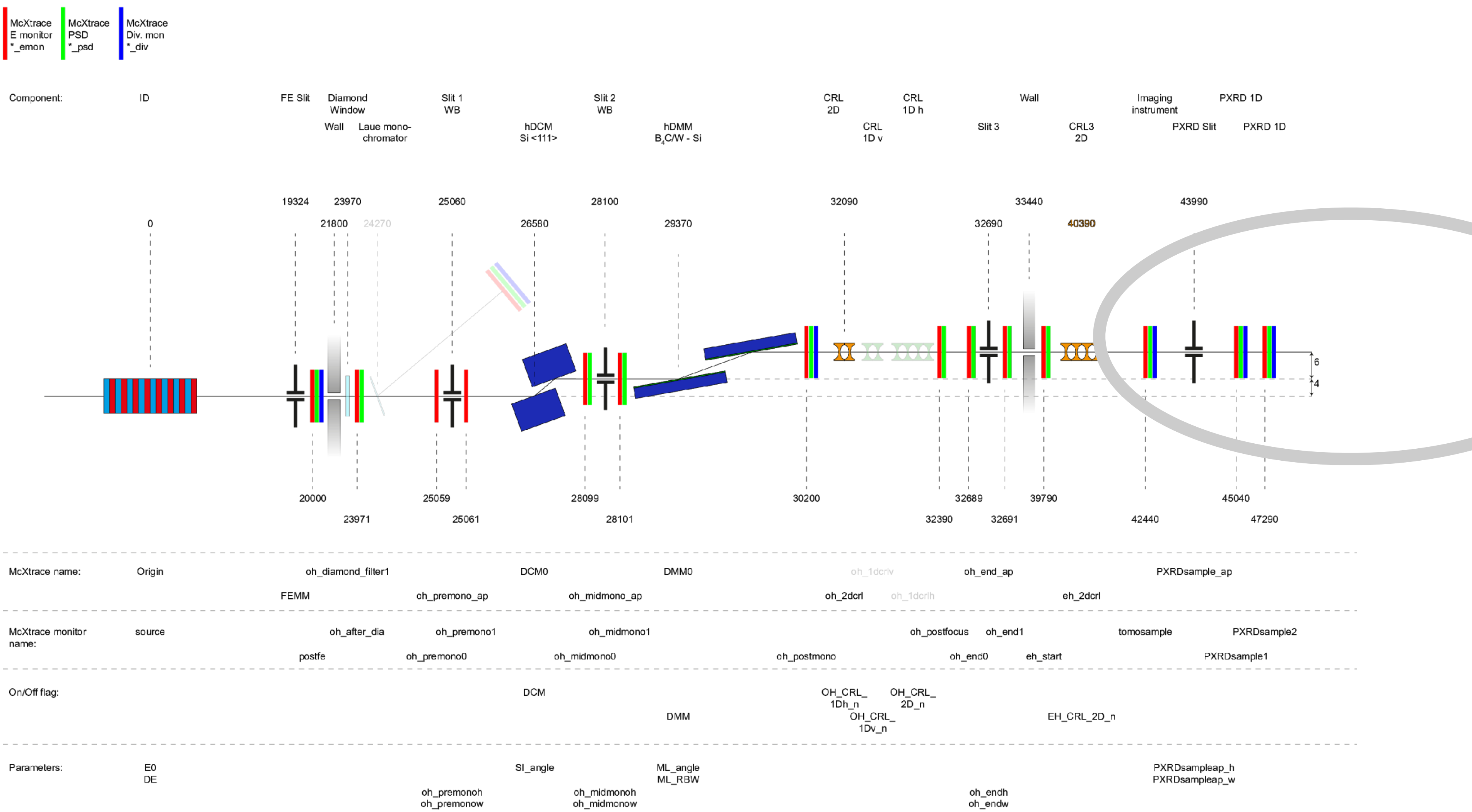


DanMAX simulation effort

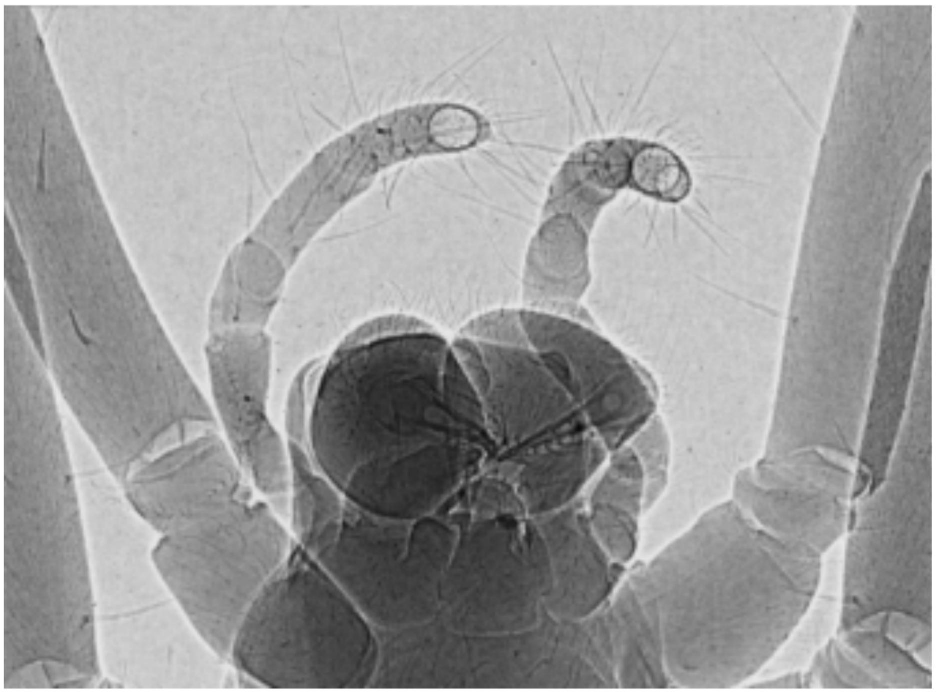
McXtrace main simulation vehicle with connections to:

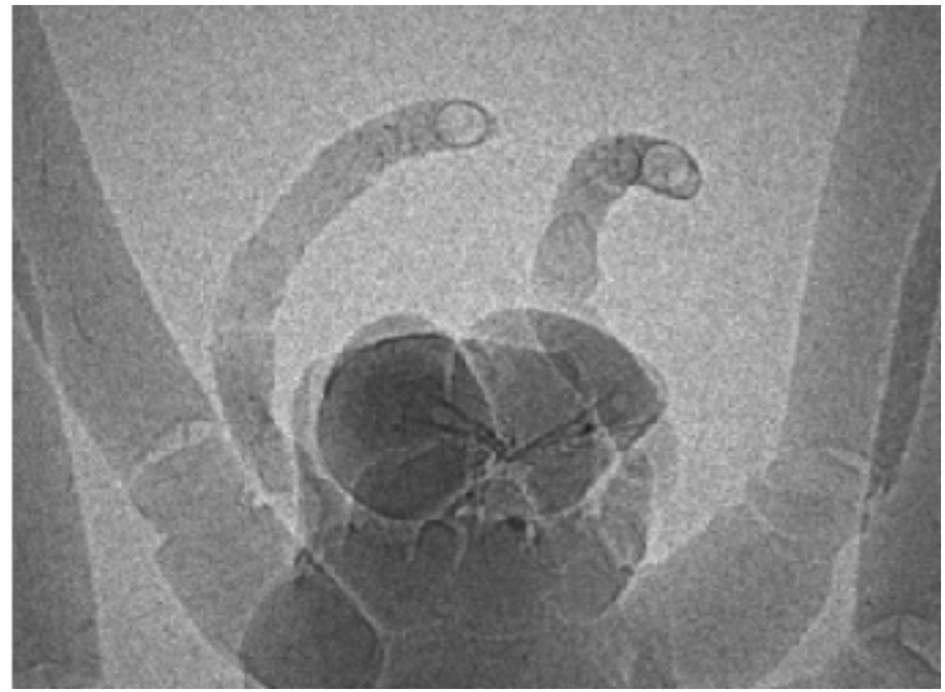
- SPECTRA / SRW for source (undulator) calculation benchmarking
- COMSOL + SHADOW + MASH – heatload calculations
- IMD – multilayer reflectivity calculations

DanMAX - McXtrace model - cheat sheet



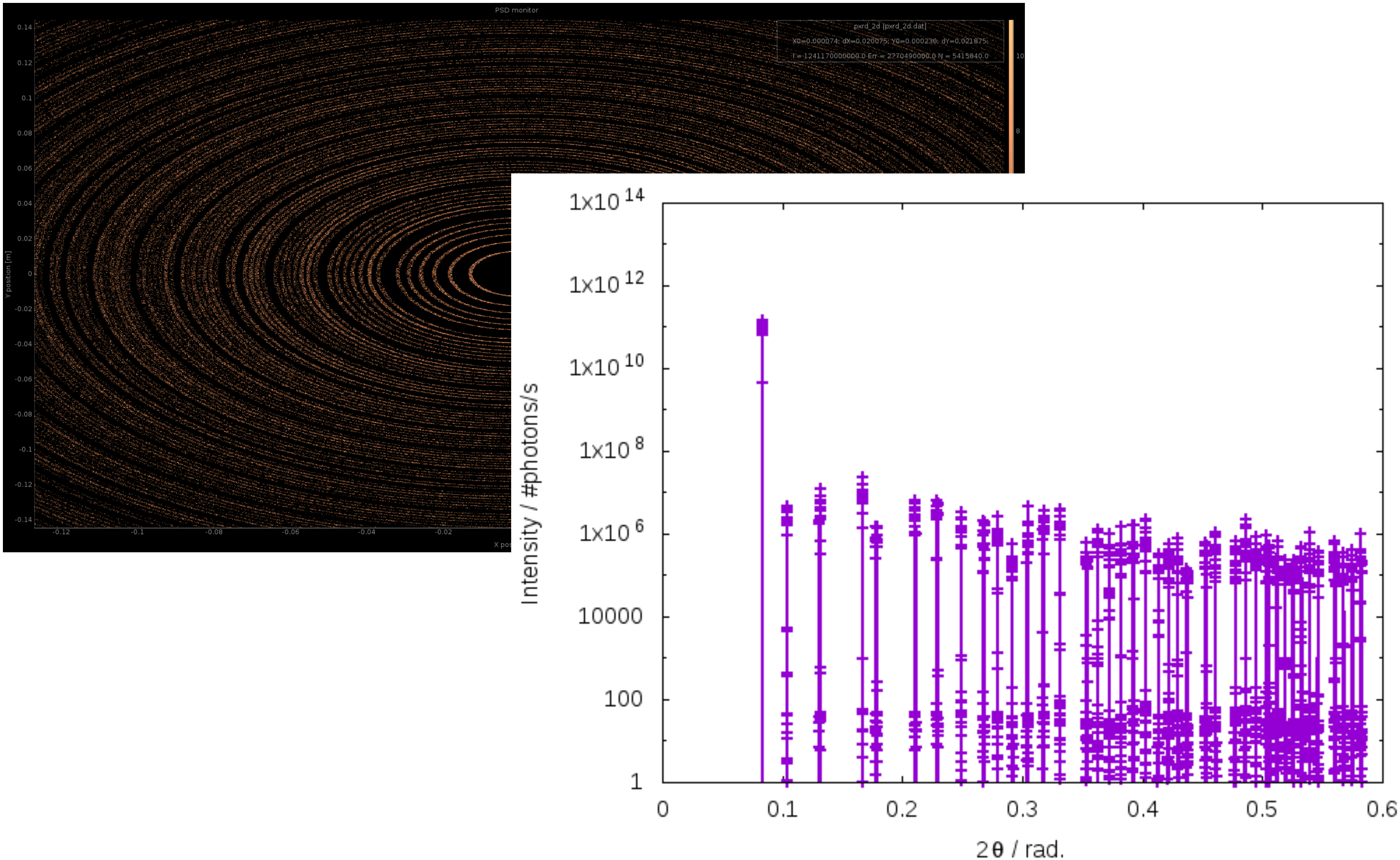
DanMAX virtual data





[I. Kantor]

DanMAX virtual data

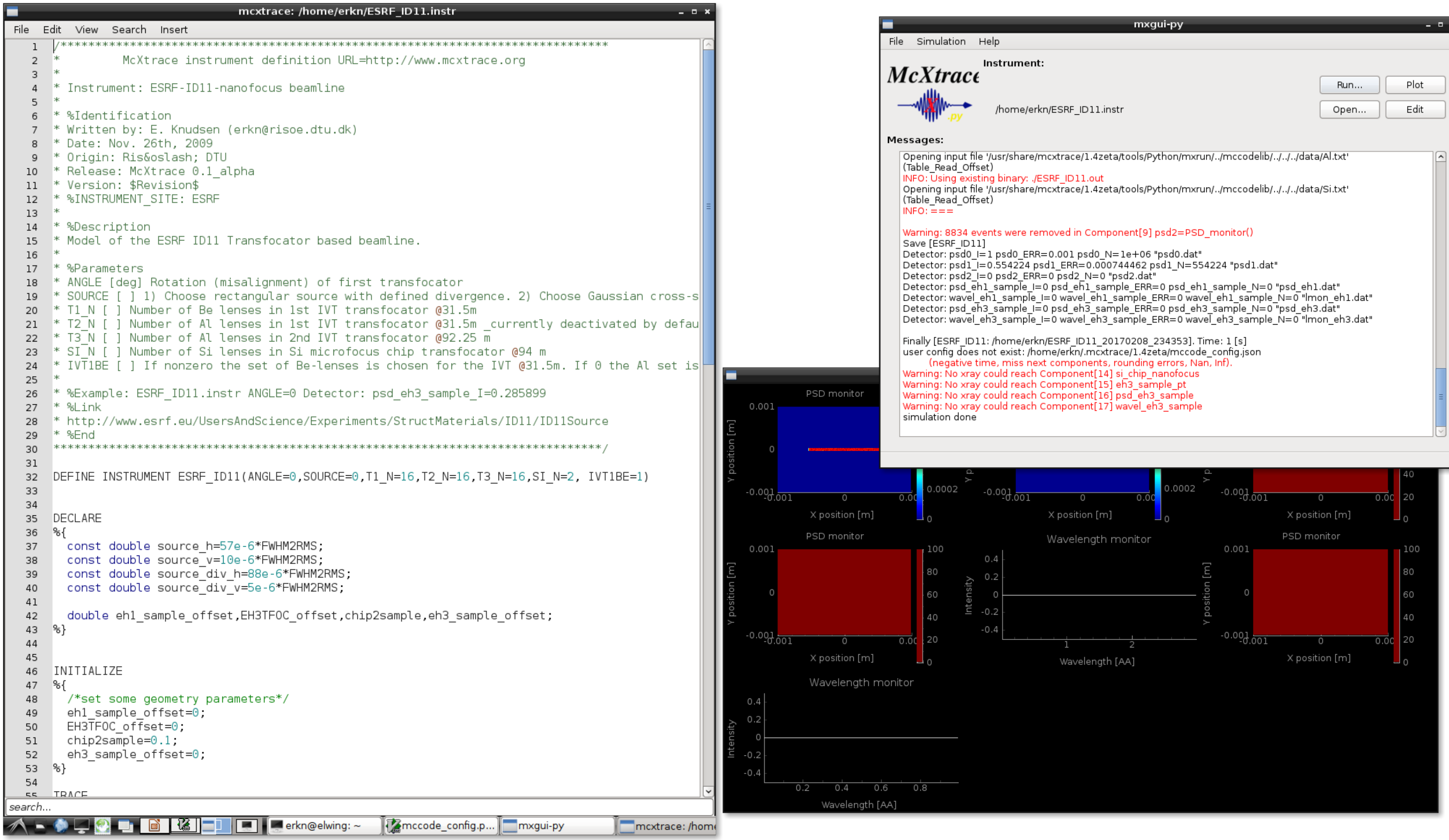


Other new features:

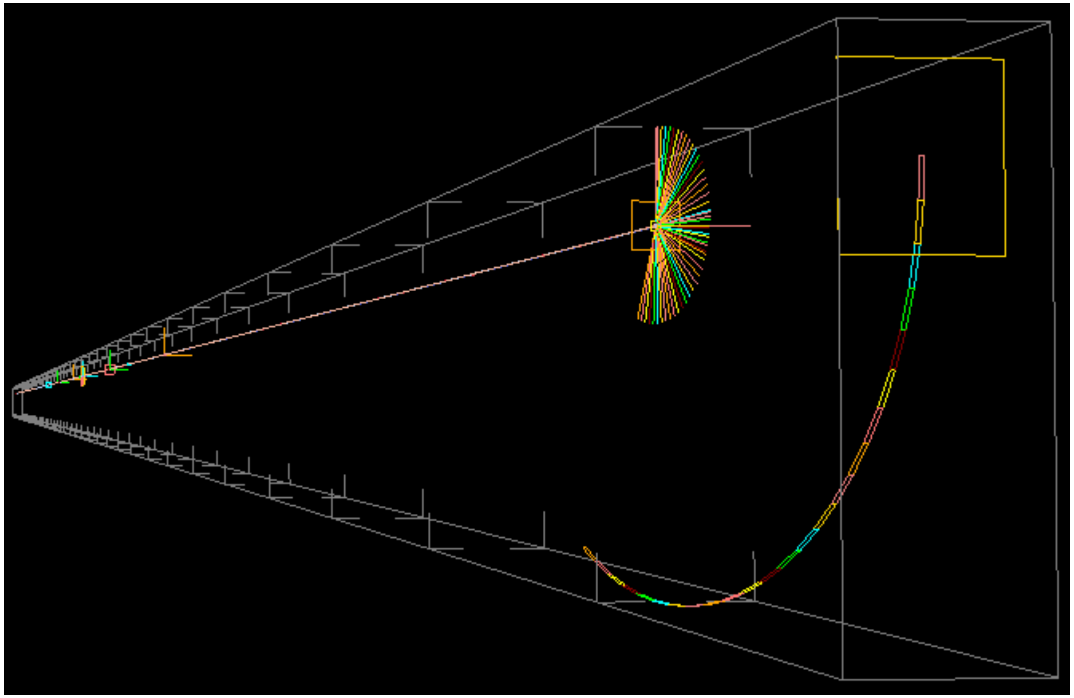
- A new set of modern, python based GUI and plotting tools
- Polycrystal sample model
- A homogenization of components of similar type, to make it easier for users to switch between models.
- More standardized installation on debian and FreeBSD class systems.
- New example simulations.
- Support for the NeXus-file format

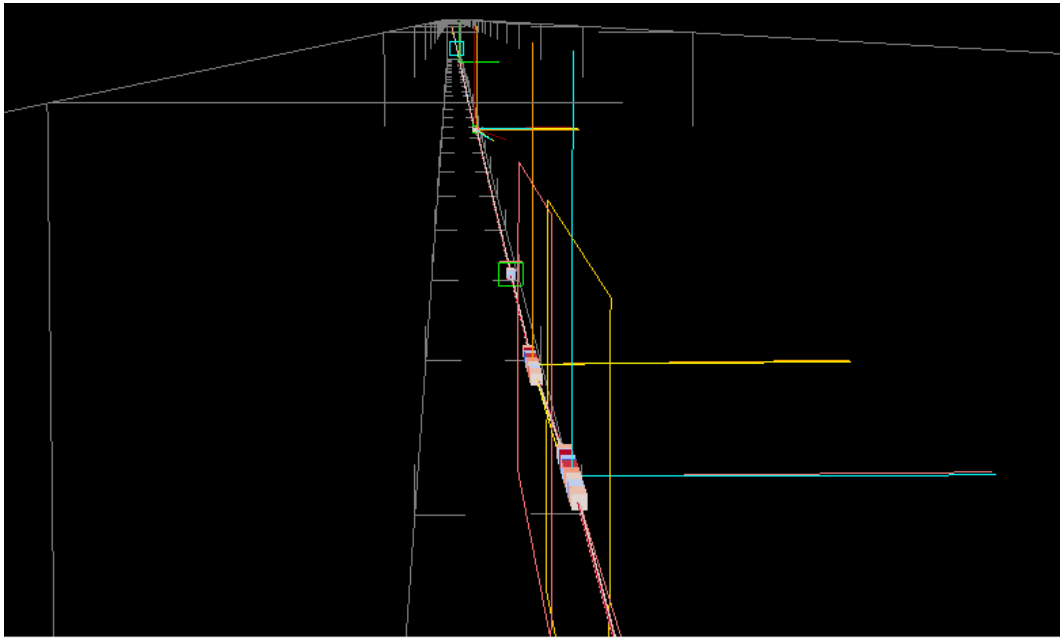
GUI

Python based



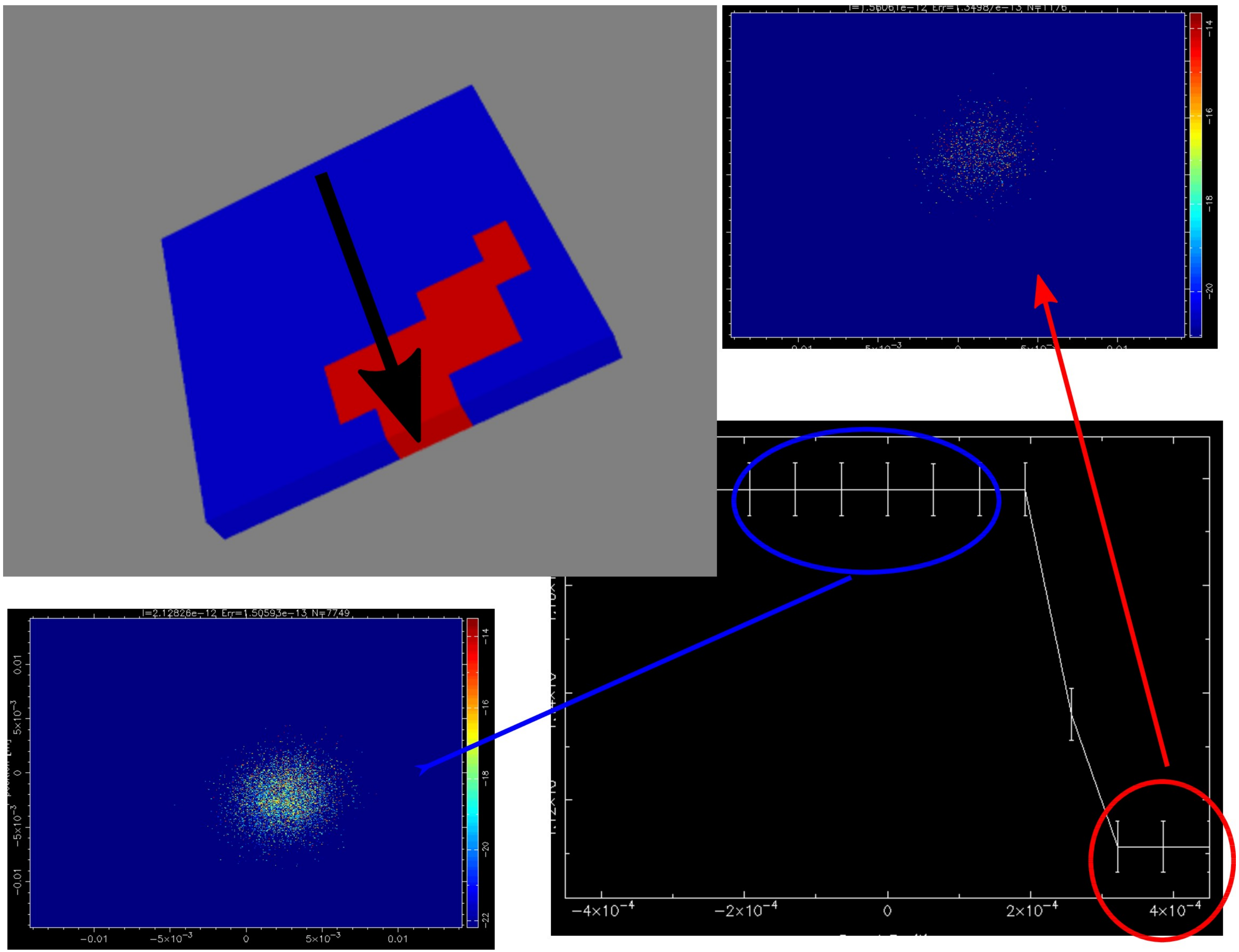
DanMAX visual ray tracing



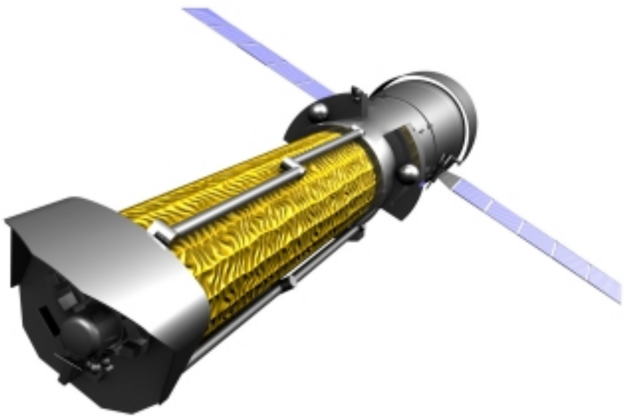


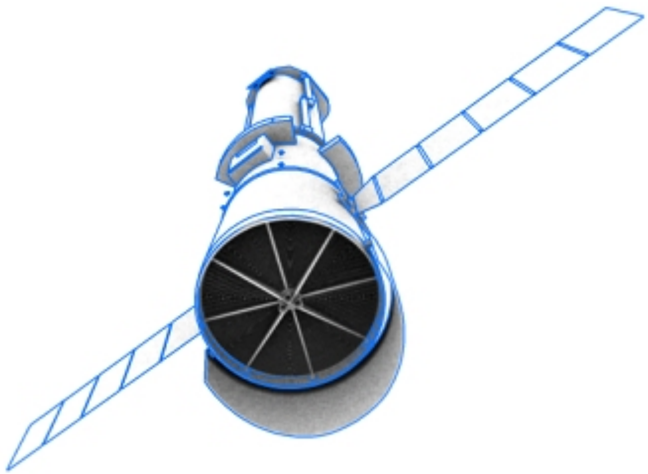
Polycrystal

Ge-bicrystal wafer scanned by a monochromatic microbeam along the arrow. The red crystal has a .7% greater lattice spacing.



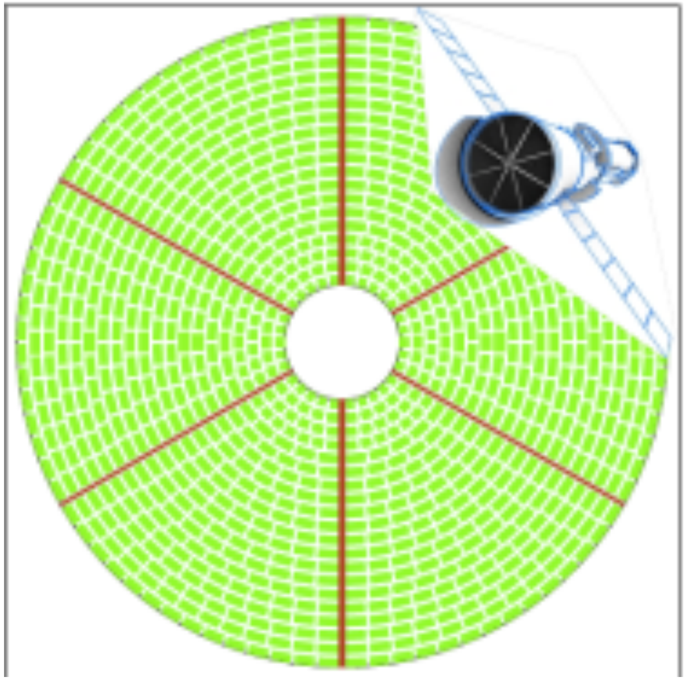
X-Ray satellite telescope

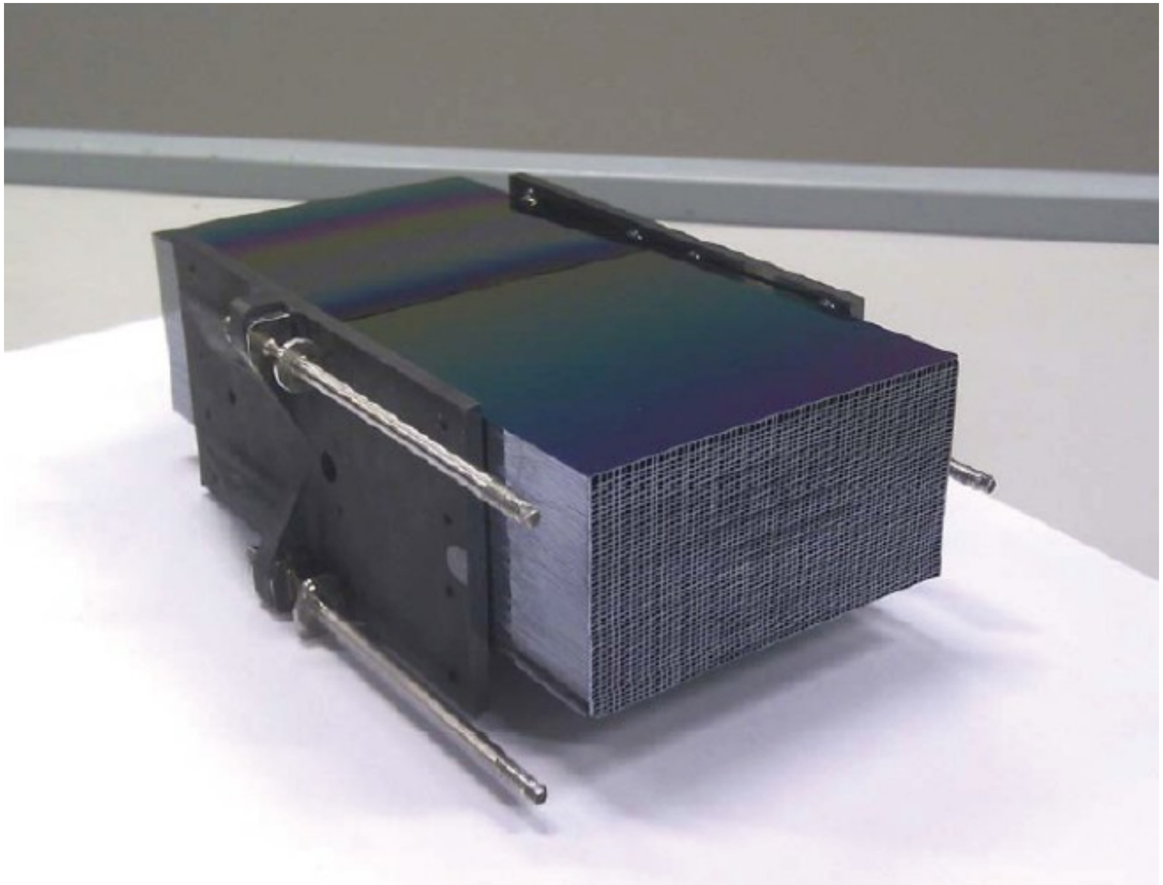




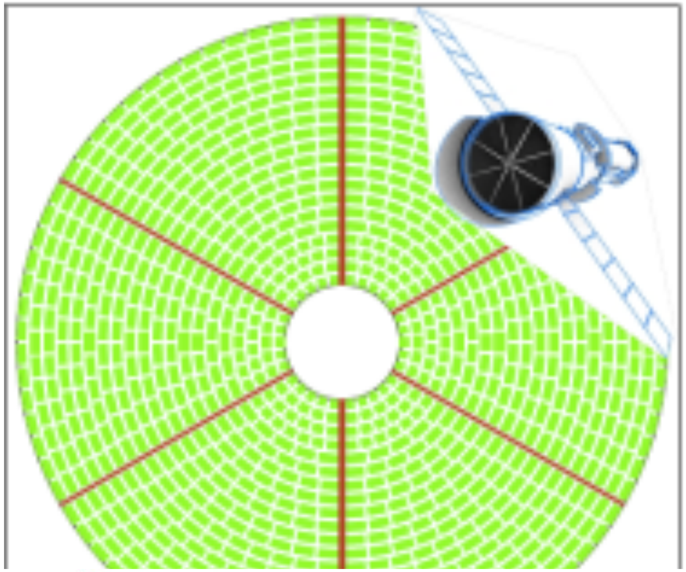
ATHENA X-ray space telescope: http://sci.esa.int/ixo/

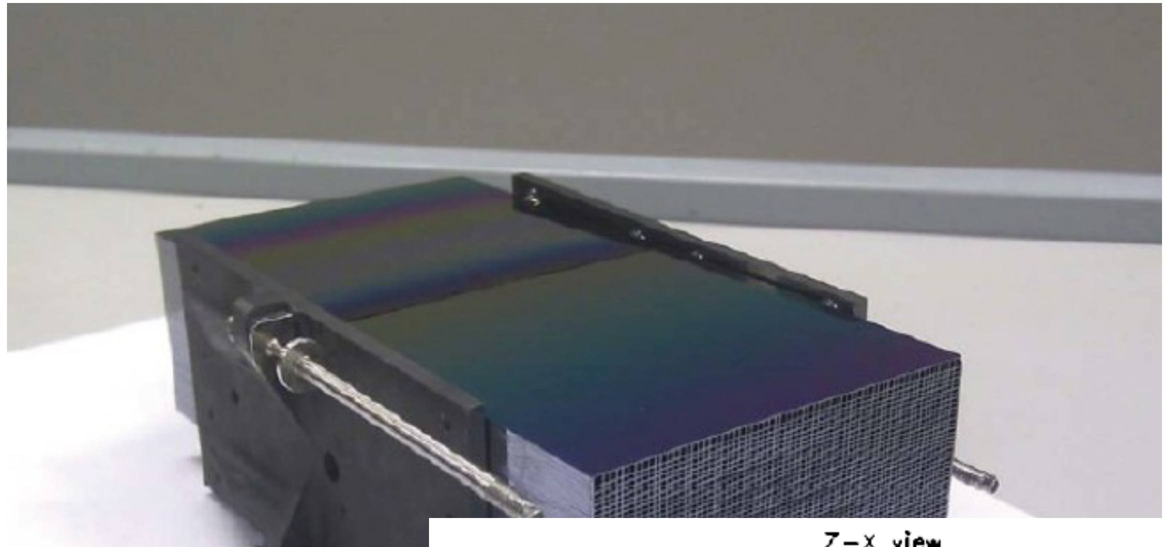
X-Ray satellite telescope

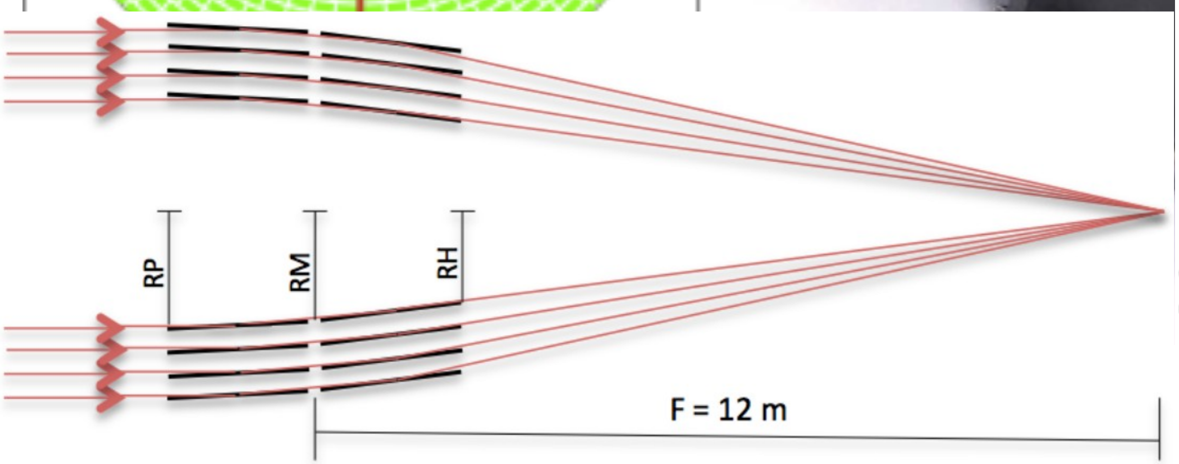


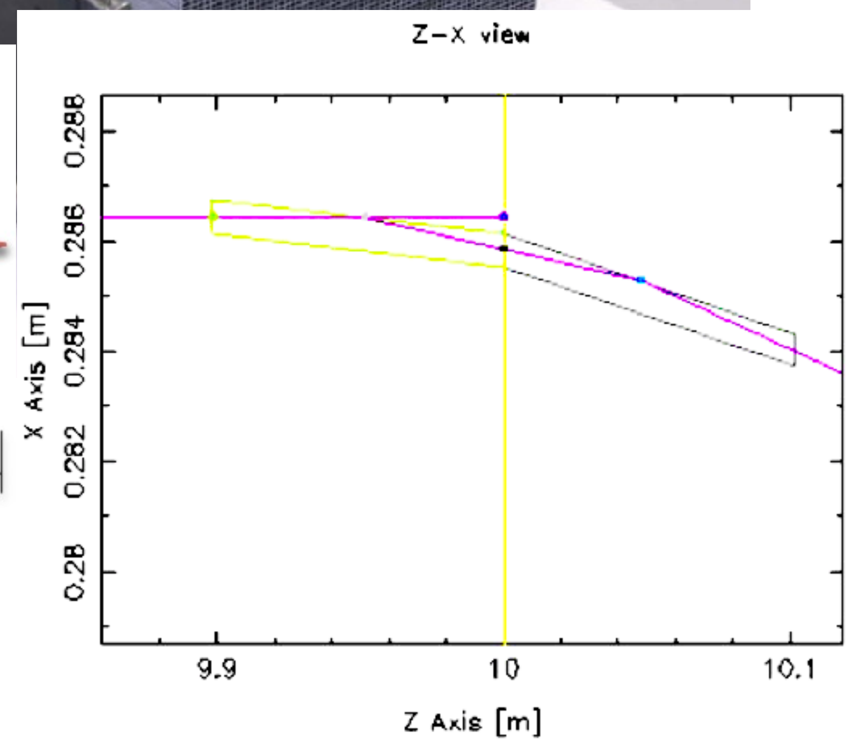


X-Ray satellite telescope

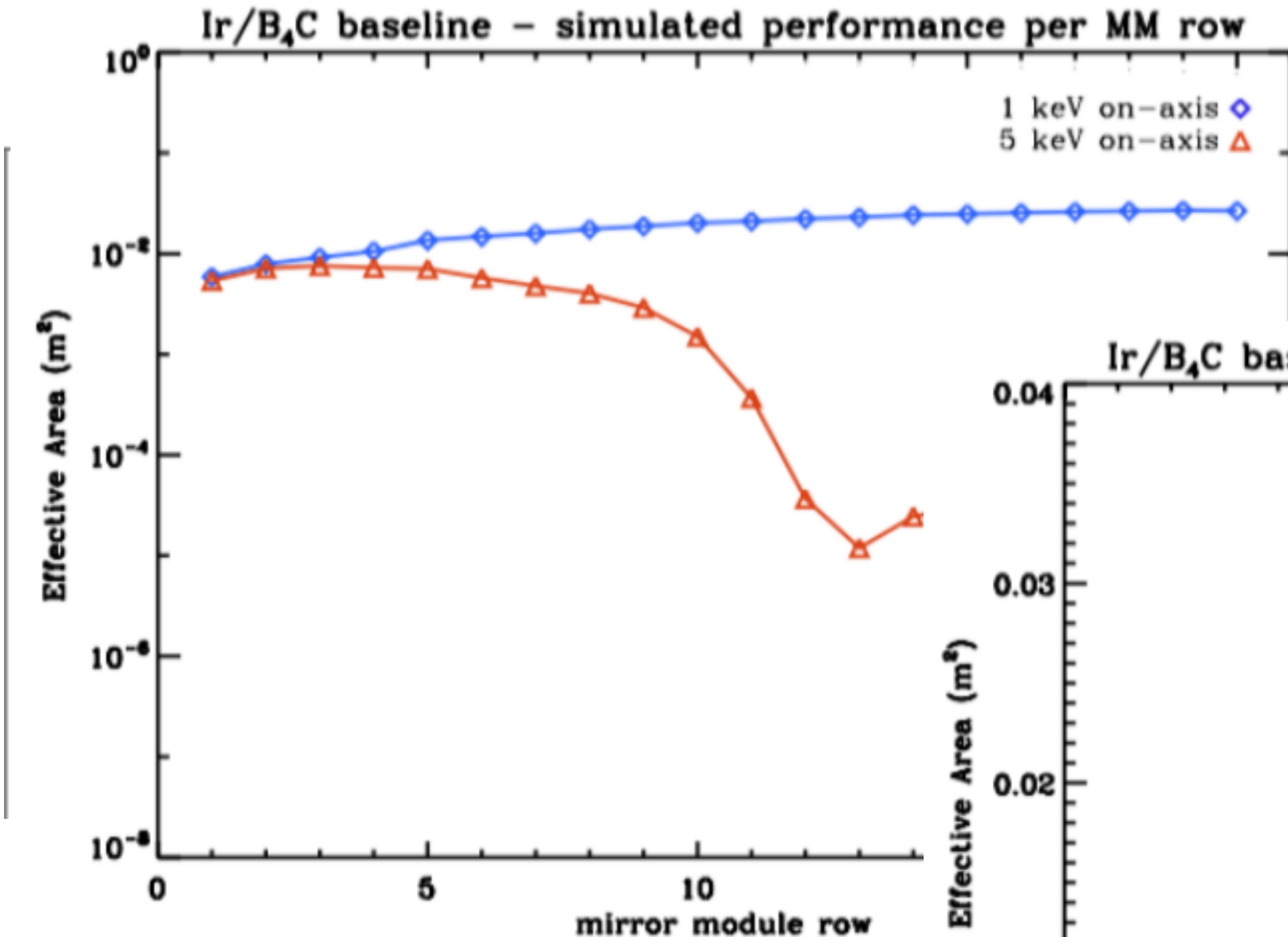


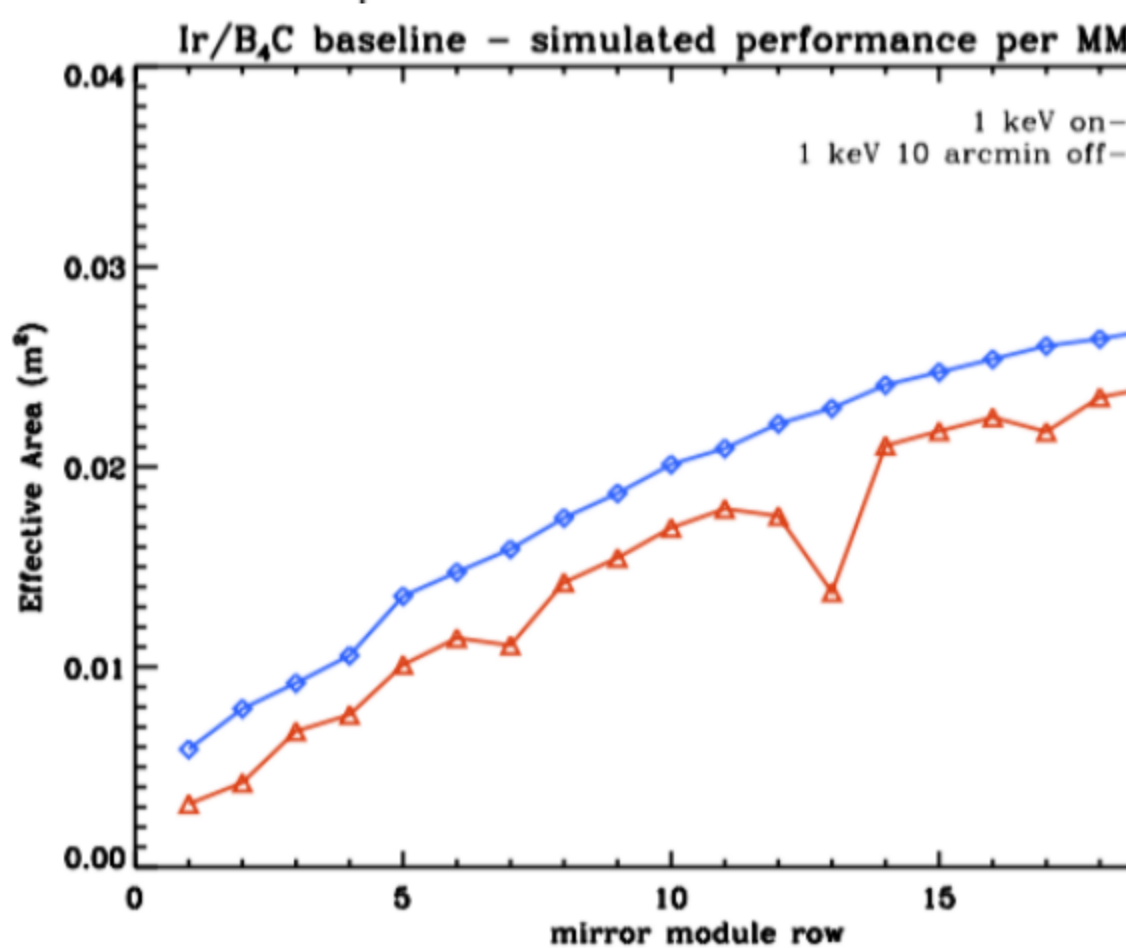






X-Ray satellite telescope





Website

McXtrace homepage - Mozilla Firefox

McXtrace homepage

mcxtrace.org

Most Visited | Getting Started | Portscout Port Up... | Getting Started | Bright: New beamlin... | Astro-Update Hea... | Portscout Port Up... | Tinkercad | Create 3... | Tændt | What Every Comput...

McXtrace - An X-ray ray-trace simulation package

- **McXtrace**
- **Project Status**
  - Project Partners
  - Project People
  - Goal
- **Installation**
  - Mac OSX Installation
  - Windows Installation
- **Mailing List**
- **Links**
- **Publications**
- **Minutes of Meetings**
- **McXtrace Art**
- **About McXtrace**
- **Documentation**
  - Commands
  - Tutorial
- **Download**
  - Components
- **Search**

McXtrace - Monte Carlo Xray Tracing, is a joint venture by

Funding from NABIIT, DSF and the above parties.

Our code is based on technology from McStas.

Code repository (shared with 'McStas') is located at github.com/McStasMcXtrace

For information on our progress, please subscribe to our user mailinglist..

- To download the latest release: download area
- For installation instructions : installation
- For a quick list of the available commands: commands

McXtrace News

The latest McXtrace news.

McXtrace 1.4: known bugs

- Lens_parab & Lens_parab_Cyl: A missing '!' in the header (line 11) trips the intersection algorithm. Fixed in the devlopment tree
- mxplot-pyqtgraph on windows: An error in a post-install script causes the file c:\mcxtrace-1.4\bin\mxplot-pyqtgraph to have a typo where it says mcplot.py instead of mxplot.py as it should. The mxplot-pyqtgraph script is generated install time - the install procedure has been altered to fix this. For existing systems the simple workaround is to either edit that file in place or to replace it with one from this link:downloads.mcxtrace.org/mcxtrace-1.4/fixes.
- On windows, if you want to use the Undulator component, you must also install The GNU scientific Library (GSL). It is available from gnuwin32.sourceforge.net/packages/gsl. This also implies that you probably need to edit the mcxtrace compiler flags. Go to File->Configure and add something along the lines "-Lc:\mcxtrace\Program Files (x86)\GnuWin32\lib -Ic:\mcxtrace\Program Files (x86)\GnuWin32\include" to the C flags box for mcxxtrace to be find the libraries compile time.

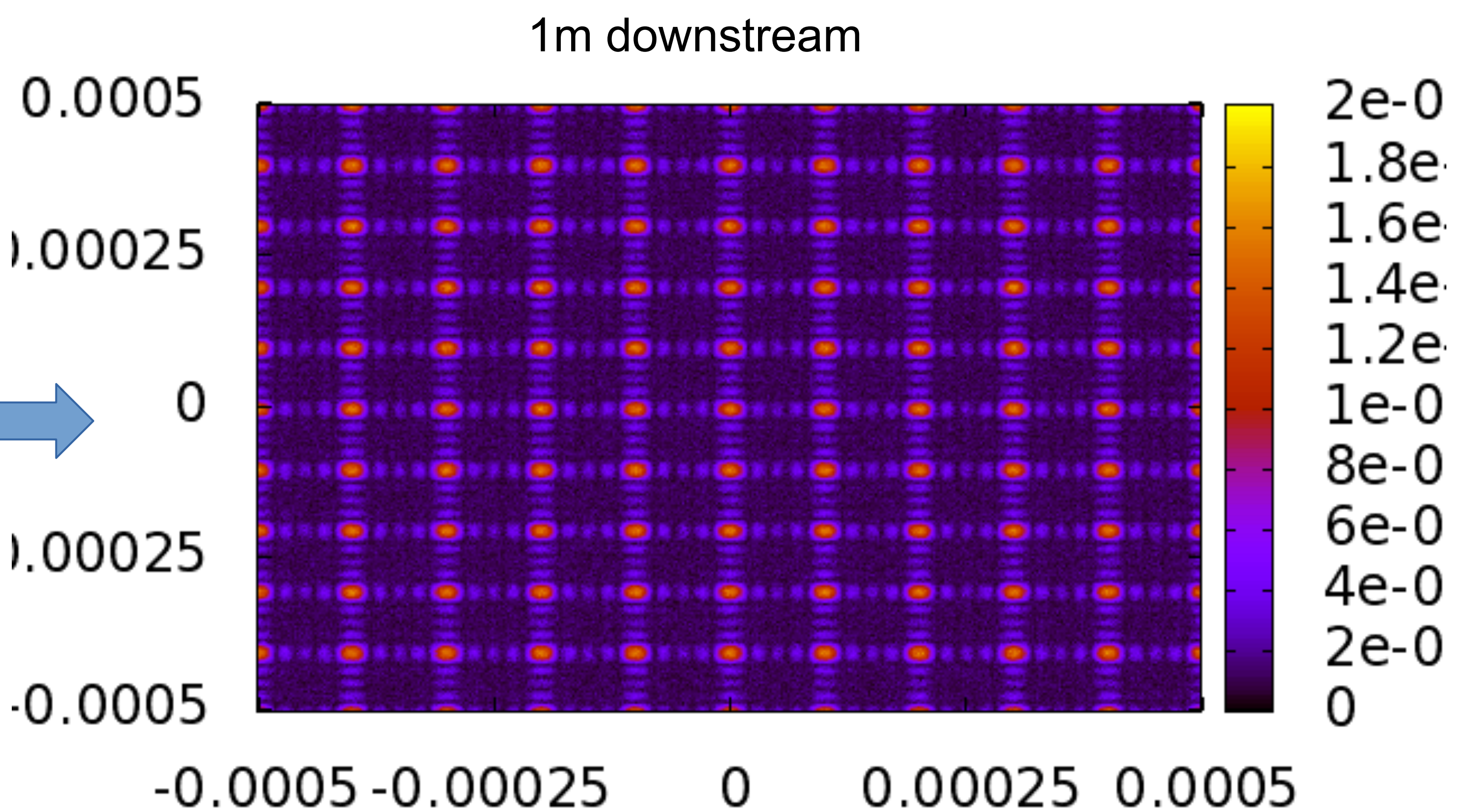
Current Projects

1. Interoperability
2. Ray Tracing Ptychography

Current Projects

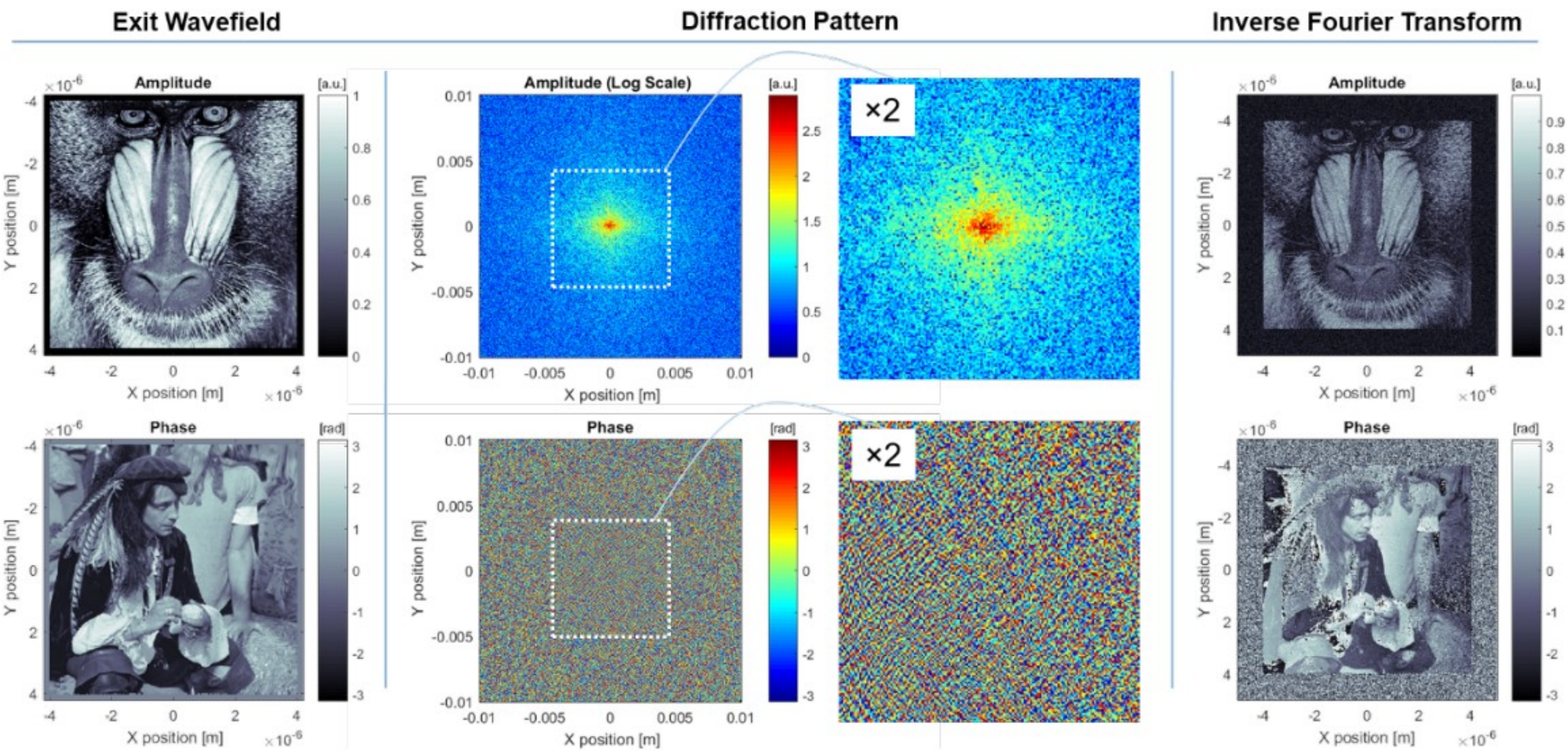
1. Interoperability
2. Ray Tracing Ptychography

5x5 Grating Point Source

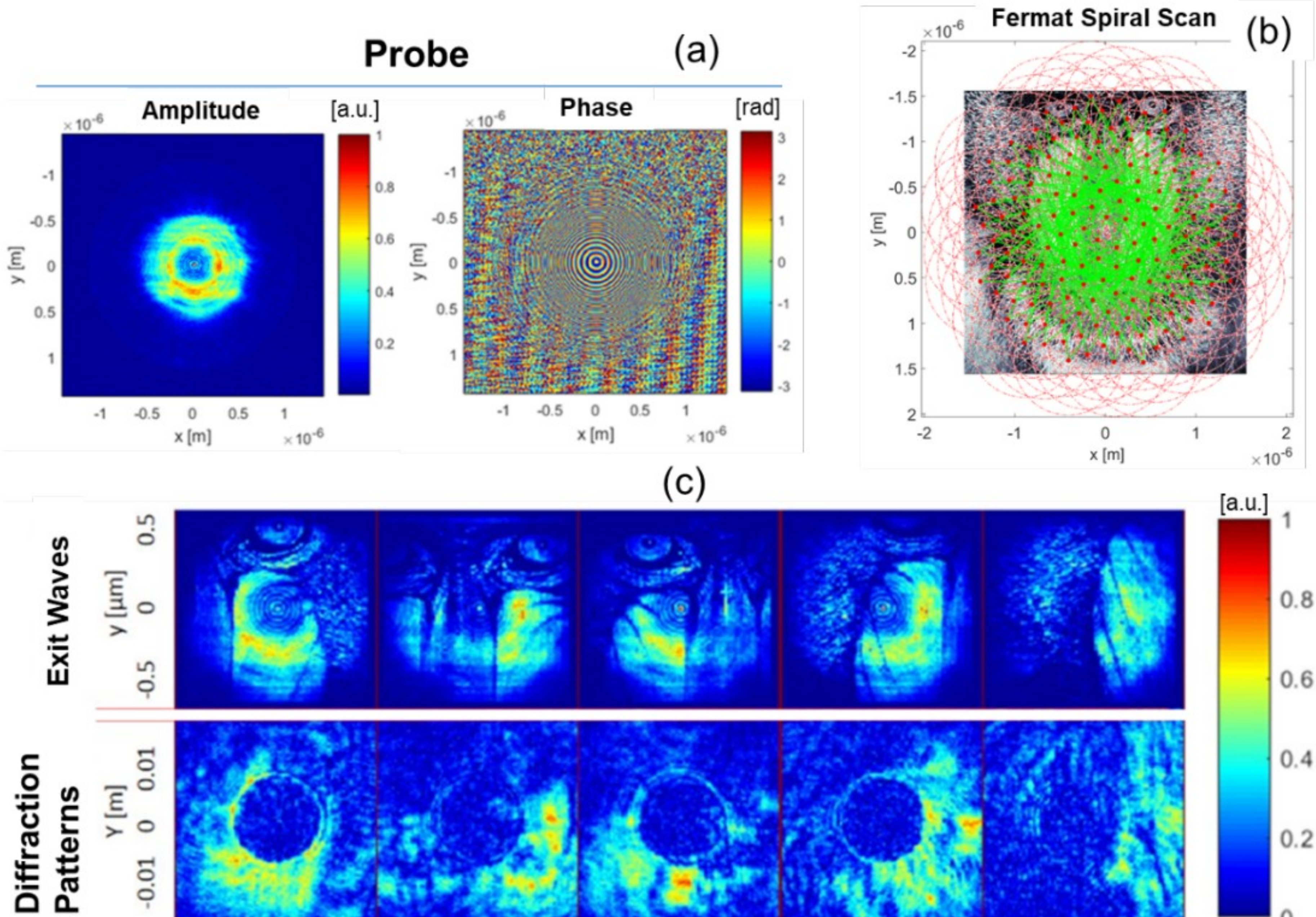


[G. Fevola]

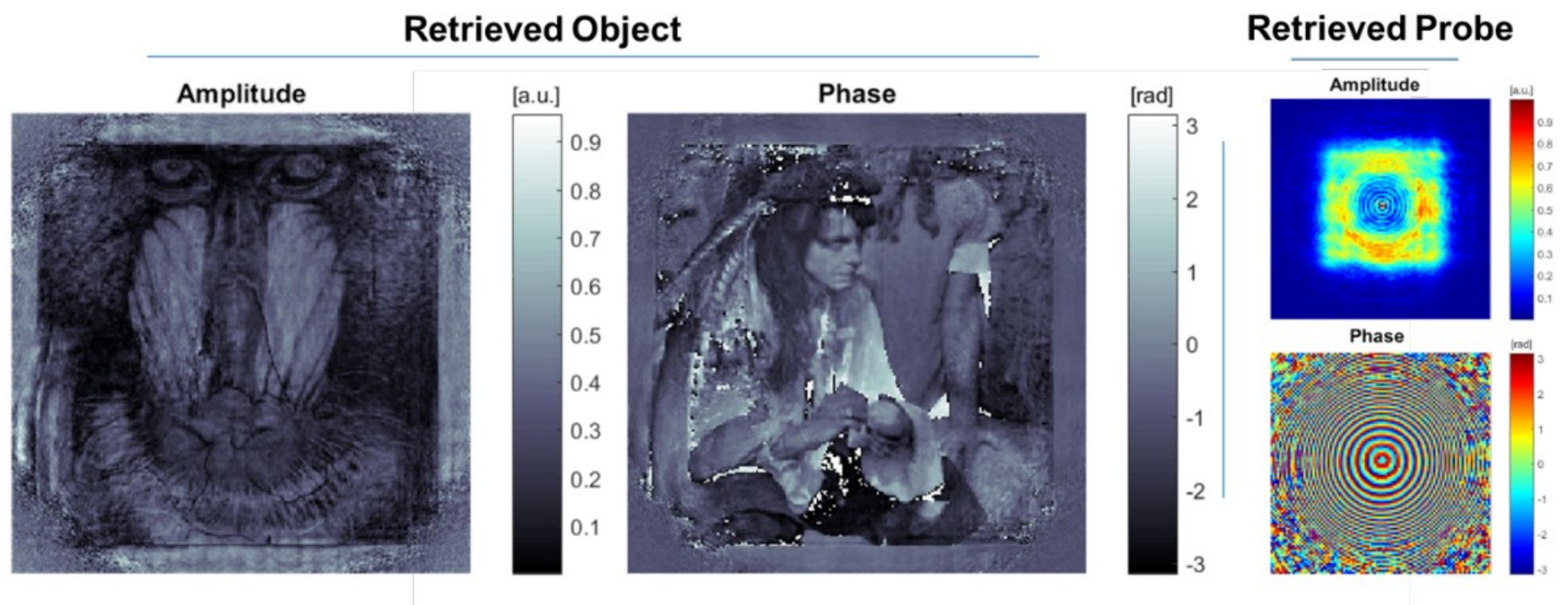
Raytracing diffraction



Raytracing Ptychography



Raytracing Ptychography



Thank you for your attention!

Peter Willendrup [1]
Andrea Prodi [4]
Jana Baltser [2]
Søren Schmidt [1]
Martin Meedom [1]
Henning Friis Poulsen [1]
Manuel Sanchez del Rio [4]
Claudio Ferrero [4]
Karsten Joensen [5]
Kell Mortensen [3]
Søren Kynde [3]
Martin Cramer Petersen [3]
Robert Feidenhans'l [2]
Kim Lefmann [2]
Matteo Busi [1]
Innokenty Kantor [1,7]
Mads Ry Jørgensen[8,7]
Marcus Mendenhall [9]
Giovannin Fevola [10]

[1] DTU Physics, Kgs. Lyngby, Denmark.
[2] Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark.
[3] Faculty of Life Science, University of Copenhagen, Copenhagen, Denmark.
[4] European Synchrotron Radiation Facility (ESRF), Grenoble, France.
[5] SAXSLAB, Denmark (Formerly JJ X-RAY Systems).
[6] European Spallation Source, Lund, Sweden.
[7] MAX IV Laboratory, Lund, Sweden.
[8] iNANO-center, Århus University, Århus, Denmark.
[9] NIST, Gaithersburg, MD, USA.
[10] DTU Energy, Roskilde, Denmark.