Erik Knudsen
Peter Willendrup
Emmanuel Farhi

And many other contributors:
Manuel Sanchez del Rio
Claudio Ferrero
Kell Mortensen
Robert Feidenshans’l
Kim Lefmann

McXtrace: X-ray Monte-Carlo ray-tracing

What is McXtrace ?

A software to simulate X-ray beam lines with incoherent and coherent sources, including optics, samples (scattering), detectors.

This is a ray-tracing code (Monte-Carlo).

Can easily couple to other codes.

Made to be extensible. Massively parallel.

"McXtrace: a Monte Carlo software package for simulating X-ray optics, beamlines and experiments",
Journal of Applied Crystallography, vol. **46** (2013) 679.

What can it be used for ?

Understand beam line as a whole.

Improve beam lines. “What if the ring lattice is changed ?”

Train users – plan experiments.

Help analyse data (limited by sample models).

How does it work

1. Describe a beam-line in the McXtrace language (text file).
2. Automatically convert beamline into ANSI ***C***
3. Compile (Automatically)
4. Run (Automatically)

→

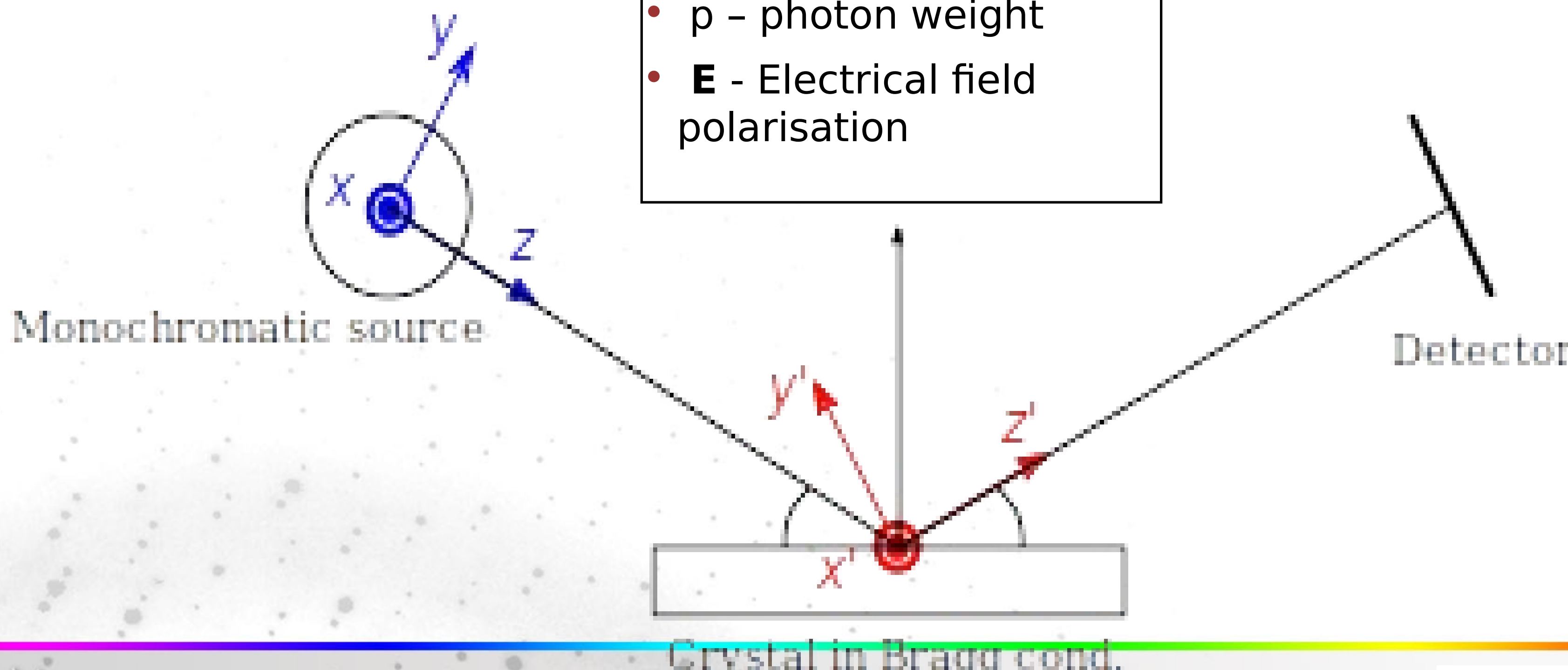
1. Optimized for your platform
2. Only includes what you use

Built over proven McStas (neutrons) technology (1998-).
Produce data sets and plots, as well as 3D views.
Typical simulation time: 10s, up to few minutes.
User interfaces are in Python (also keep legacy Perl). Matlab interface.

A photon particle is...

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



Installation

Available on all systems (Windows, Linux, Mac) at <http://www.mcxtrace.org/>.
[1.5 RC1: **http://downloads.mcxtrace.org/mcxtrace-1.5rc1/**]

Available as Virtual Machine <http://ord03246.synchrotron-soleil.fr> (exp.)

Available example at <https://sim.e-neutrons.org>

Help / Community

Extensive manuals, tutorials, examples.

Forum/mailing list.

Built over McStas – vast knowledge, community and expertise.

Code hosted at <https://github.com/McStasMcXtrace/McCode>

What you get in the box

A software that simply runs...

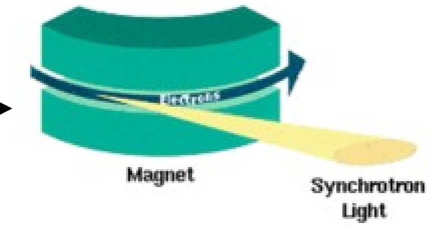
- **Components (80)**
  - Sources
  - Optics
  - Samples
  - Detectors
- **A GUI to assemble and run BL.**
- **Tools to view the results (*e.g.* detectors).**
- **Tools to view the BL geometry.**
- **Examples (40)**
- **Extensive documentation, mailing list...**

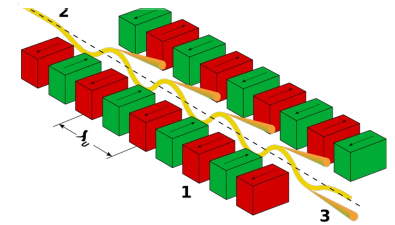
User interfaces *and* scripting (bash, Python, Matlab).
Single shot and parameter-scanned simulations.
Generated data files as text or HDF5/NeXus.
Fully adapted to HPC (MPI).
Optimisation (swarm, GA, simplex, Hooke, Powell, …) via ***iFit*** <http://ifit.mccode.org>

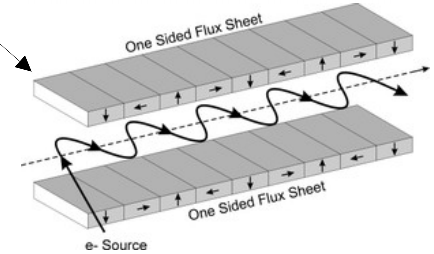
Sources

Synchrotron ID

- Bending magnet [B.D. Patterson, *Am. J. Phys.* **79**, 1046 (2011)]
- Undulator [K.J. Kim, *AIP, conf. proc.*, 184, 1989]
- Wiggler [B.D. Patterson, *Am. J. Phys.* **79**, 1046 (2011)]







Lab/ideal stuff

- Laboratory X-ray tube (e.g. rotating anode)
- Ideal, point and gaussian

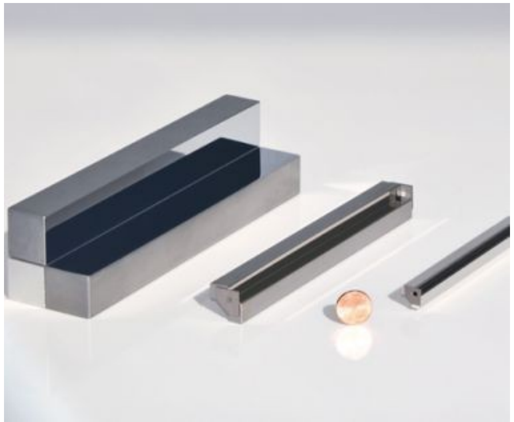
Interfaces with other software

- ***Spectra*** (R) <http://spectrax.org/spectra/>
- ***Simplex*** (R) <http://spectrax.org/simplex/index.html >
- ***Genesis*** (R) <http://genesis.web.psi.ch/>
- ***Shadow*** (RW) <https://github.com/oasys-kit/shadow3>
- ***MCPL*** (GEANT4, PHITS, MCNP) (RW) <https://mctools.github.io/mcpl/>

Optics

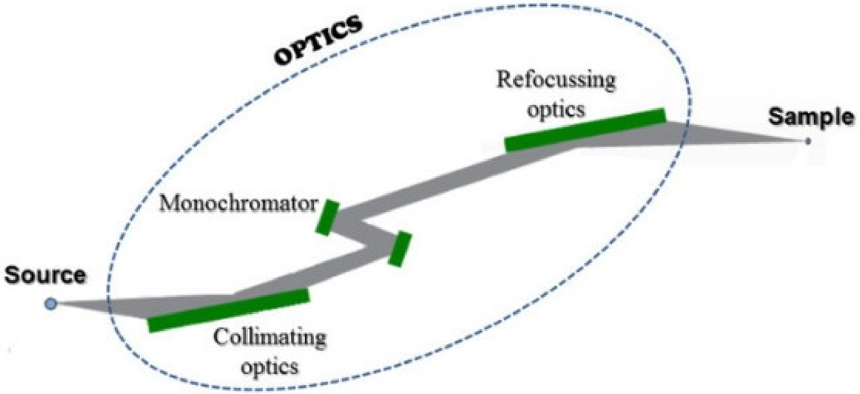
Enough to start with:

- Bragg crystal (monochromator, incl. bent)
- Capillary
- Filter (absorption and refraction)
- Lenses
- Mirrors (flat, curved, multi-layers, twin KB multi-layer)
- Zone plate
- Grating (lamellar, blazed)
- Slit, beam-stop, ...



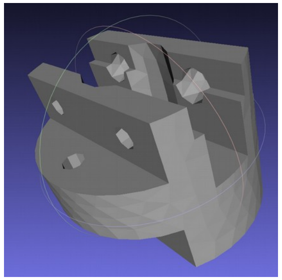
More to come (with your help ?)

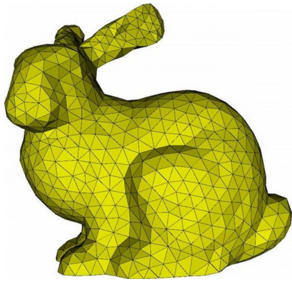
Components can be arranged in groups.



Samples

- SAXS sample (60 models from SasView, PDB, Nanodiscs, Liposomes, *I(q)*, …)
- Powder (diffraction)
- Polycrystal (diffraction)
- Pump-probe (2 states) molecule
- Single crystal (diffraction – hold on !)





All samples can have simple geometric shapes (incl. hollow).
Powder and SX can have any shape (PLY/STL).
Powder sample supports multiple concentric geometries (*e.g.* for cryostat).
McXtrace comes with a material data base, and can use *e.g.* NIST files.

Detectors

Currently only record distributions (1D, 2D, event lists, …) for any combination of state parameters (position, divergence, energy, power, phase, E-field, …).

Not satisfactory !

Actual detector efficiency to be added (simple models exist), e.g.

- <https://bl831.als.lbl.gov/~jamesh/mlfsom/>

Examples

SAXS

Powder

Single crystal

Coherence

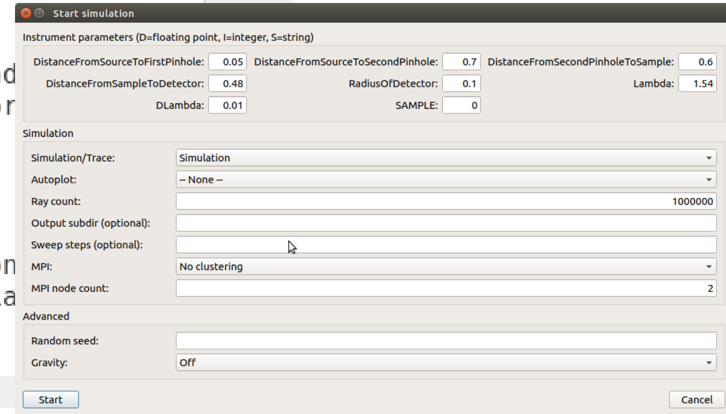
MaxII 711 XRD/MX

DanMAX@MaxIV Powder XRD 2D
https://sim.e-neutrons.org/instrument-menu/xrd/MAXIV_DanMAX_pxrd2d

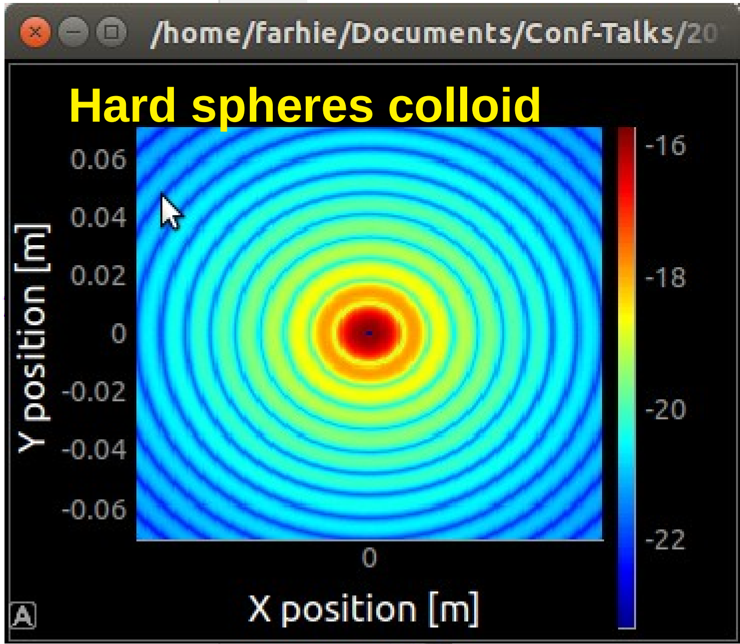
SAXS

McXtrace

```
/*******************************************************************************
*         McXtrace instrument definition URL=http://www.mcxtrace.org
*
* Instrument: TestSAXS
*******************************************************************************/

DEFINE INSTRUMENT TestSAXS(
  DistanceFromSourceToFirstPinhole = 0.05, DistanceFromSourceToSecond
DistanceFromSecondPinholeToSample = 0.6, DistanceFromSampleToDetector
RadiusOfDetector = 0.1, Lambda = 1.54, DLambda = 0.01, SAMPLE=0 )

TRACE
COMPONENT Source = Source_flat(
  xwidth = 0.0005, yheight = 0.0005, dist = DistanceFromSourceToSecon
focus_yh = 0.0003, focus_xw = 0.0003, lambda0 = Lambda, dlambda = DLa
AT (0, 0, 0) ABSOLUTE

COMPONENT FirstPinhole = Slit(radius = 0.0002)
AT (0, 0, DistanceFromSourceToFirstPinhole) RELATIVE Source

COMPONENT SecondPinhole = Slit(radius = 0.00015)
AT (0, 0, DistanceFromSourceToSecondPinhole) RELATIVE Source

COMPONENT Sample0 = SAXSSpheres(
  xwidth = 0.01, yheight = 0.01, zdepth = 0.01, R = 50.0, SampleToDetectorDistance =
DistanceFromSampleToDetector, DetectorRadius = RadiusOfDetector )
AT(0, 0, DistanceFromSampleToDetector) RELATIVE SecondPinhole

COMPONENT Beamstop = Beamstop(
  radius = 0.001)
AT (0, 0, DistanceFromSampleToDetector - 0.000001) RELATIVE Sample0

COMPONENT Monitor = PSD_monitor(
  filename = "PSDMonitor", xwidth = 2.0 * RadiusOfDetector / sqrt(2
* RadiusOfDetector / sqrt(2.0), nx = 200, ny = 200, restore_xray =
AT (0, 0, 0.000001) RELATIVE Beamstop

COMPONENT QMonitor = SAXSQMonitor(
  RadiusDetector = RadiusOfDetector, DistanceFromSample =
DistanceFromSampleToDetector, LambdaMin = Lambda, Lambda0 = Lambda,
2000 )
AT (0, 0, 0.0000010000000001) RELATIVE Beamstop

END
```

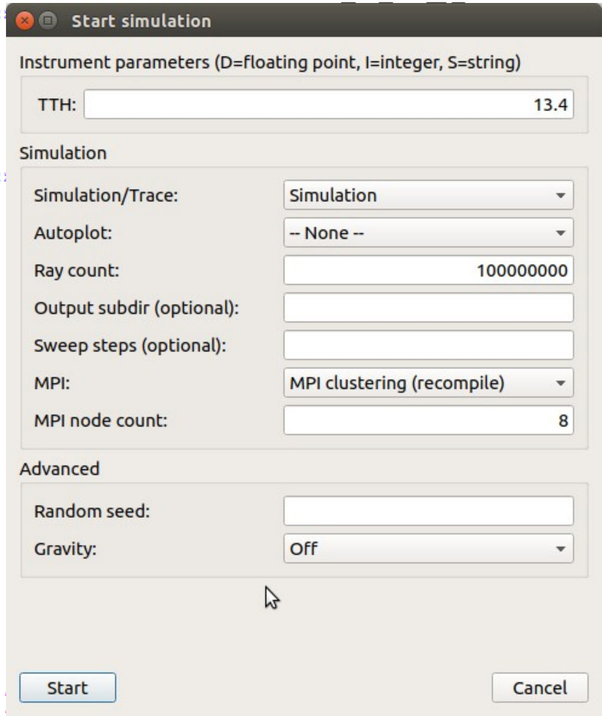


11 s

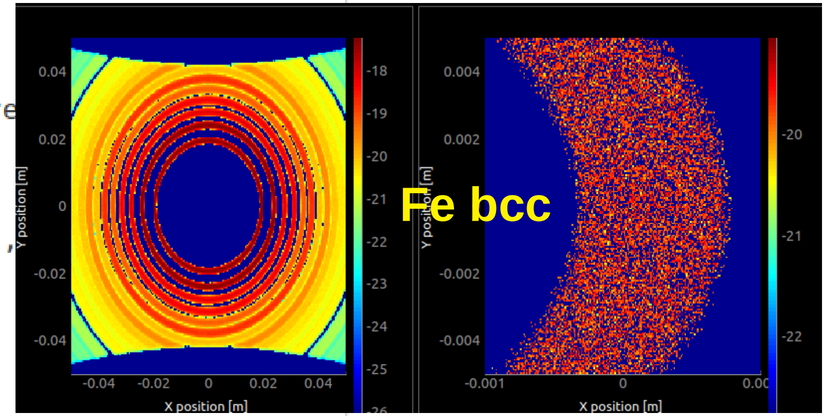


Powder Diff.

```
/*******************************************************************************
* Instrument: Test_PowderN
* %Parameters
* TTH: [deg.] Two theta rotation for detector arm
* %End
*******************************************************************************

DEFINE INSTRUMENT Test_PowderN(TTH=13.4)

TRACE
COMPONENT src = Source_flat(
    yheight = 1e-3, xwidth = 1e-3, dist = 10, focus_xw = 1e-3,
    focus_yh = 1e-3, E0 = 45, dE = 1)
  AT (0, 0, 0) ABSOLUTE

COMPONENT psd0 = PSD_monitor(
    nx = 100, ny = 100, filename = "psd0", xwidth = 2e-3, yheight = 2e-3)
  AT (0, 0, 1e-9) RELATIVE src

COMPONENT sample = PowderN(
    reflections = "Fe_bcc229_lt13deg.dat", format=Crystallographica,
    material = "Fe.txt", radius = .5e-4, yheight = 1e-3, pack = 0.5, Vc = 
    p_inc = 0, p_transmit = 0.1, DW = 0, d_phi = 45)
  AT (0, 0, 10) RELATIVE PREVIOUS
EXTEND %{
  if(!SCATTERED) ABSORB;
%}

COMPONENT psd4pi = PSD_monitor_4PI(
    nx = 180, ny = 180, filename = "psd4pi", radius = 0.1, re
  AT (0, 0, 0) RELATIVE sample

COMPONENT detector = PSD_monitor(
    nx=200, ny=200, xwidth=0.1, yheight=0.1, filename="psd1",
  AT(0,0,0.1) RELATIVE sample

COMPONENT ttharm= Arm()
  AT(0,0,0) RELATIVE sample
  ROTATED (0,TTH,0) RELATIVE sample

COMPONENT detector2 = PSD_monitor(
    nx=200, ny=200, xwidth=2e-3, yheight=1e-2, filename="psd2",restore_xray=1)
  AT(0,0,0.1) RELATIVE ttharm
END
```

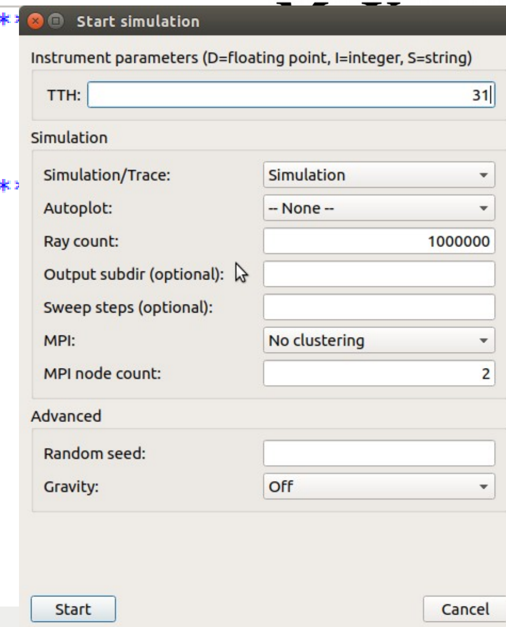


6 s

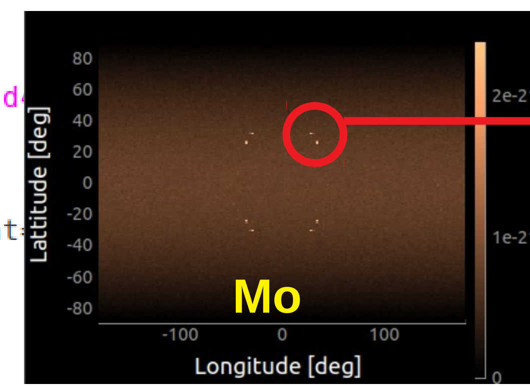


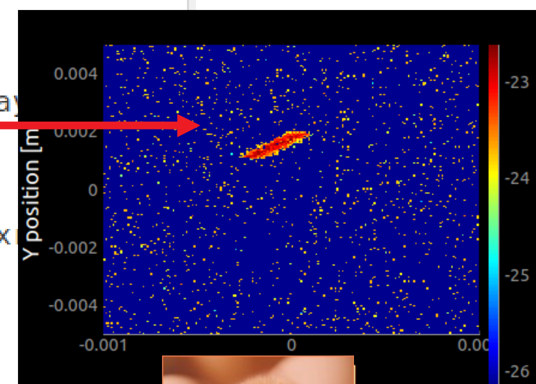
Single crystal diff.

```
/*****************************************************************************
* Instrument: Test_SX
* %Parameters
* TTH: [deg.] Two theta rotation for detector arm
* %End
*****************************************************************************

DEFINE INSTRUMENT Test_SX(TTH=13.4)

TRACE
COMPONENT src = Source_flat(
    yheight = 1e-3, xwidth = 1e-3, dist = 10, focus_xw = 1e-3,
    focus_yh = 1e-3, E0 = 45, dE = 1)
  AT (0, 0, 0) ABSOLUTE

COMPONENT psd0 = PSD_monitor(
    nx = 100, ny = 100, filename = "psd0", xwidth = 2e-3, yheight = 2e-3)
  AT (0, 0, 1e-9) RELATIVE src

COMPONENT sample = Single_crystal(reflections="Mo.lau", material="Mo.txt",
  radius = .5e-4, yheight = 1e-3, p_transmit = 0.01, mosaic=1)
  AT (0, 0, 10) RELATIVE PREVIOUS
  ROTATED (0,0,0) RELATIVE PREVIOUS
EXTEND %{
  if(!SCATTERED) ABSORB;
%}

COMPONENT psd4pi = PSD_monitor_4PI(
    nx = 180, ny = 180, filename = "psd
  AT (0, 0, 0) RELATIVE sample

COMPONENT detector = PSD_monitor(
    nx=200, ny=200, xwidth=0.1, yheight
  AT(0,0,0.1) RELATIVE sample

COMPONENT ttharm= Arm()
  AT(0,0,0) RELATIVE sample
  ROTATED (TTH,TTH,0) RELATIVE sample

COMPONENT detector2 = PSD_monitor(
    nx=200, ny=200, xwidth=2e-3, yheight=1e-2, filename="psd2",restore_xray=1)
  AT(0,0,0.1) RELATIVE ttharm
END
```



32 s



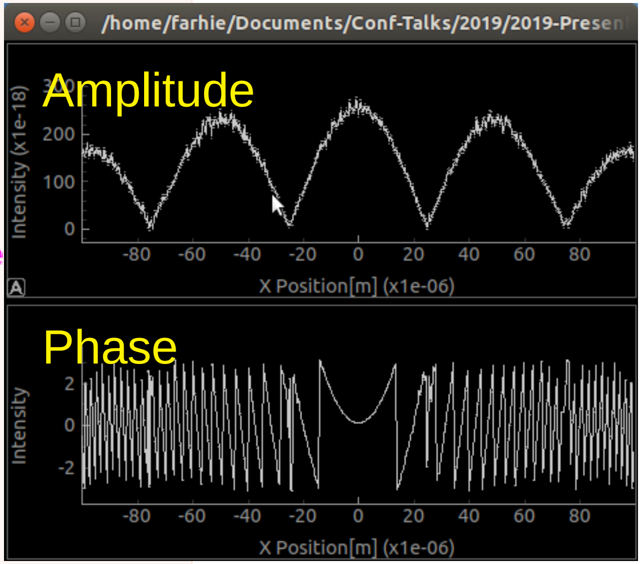


Very early stage !

2 Slits interferences

```
/*******************************************************************************
* Instrument: template_2Slit_Diff
* An example instrument showing a Young's double slit experiment
* %Parameters
* SLITW: [m] Width of the slit in question. The
* %End
*******************************************************************************/

DEFINE INSTRUMENT template_2Slit_Diff(SLITW=1e-6, SLITSEP=2e-6

DECLARE %{
  double slit_offset;
%}

INITIALIZE %{
  slit_offset=SLITSEP*0.5;
%}

TRACE

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

COMPONENT source=Source_pt(focus_xw=6e-6, focus_yh=0.8e-6, dist=1,
  lambda0=2.0, dE=0, gauss=1, randomphase=0)
AT(0,0,0) RELATIVE Origin

COMPONENT s1 = Slit(
    xwidth=SLITW, yheight=0.8e-6, dist=1, focus_xw=200e-6,focus_yh=2e
  focus_x0=0.0,focus_y0=0.0 )
AT(slit_offset,0,1) RELATIVE source
GROUP slits

COMPONENT s2 = COPY(s1)()
AT(-slit_offset,0,1) RELATIVE source
GROUP slits

COMPONENT psd0 = PSD_monitor_coh(
    yheight=2e-6, xwidth=200e-6, nx=501, ny=1, filename="psd0")
AT(0,0,2) RELATIVE source

END
```

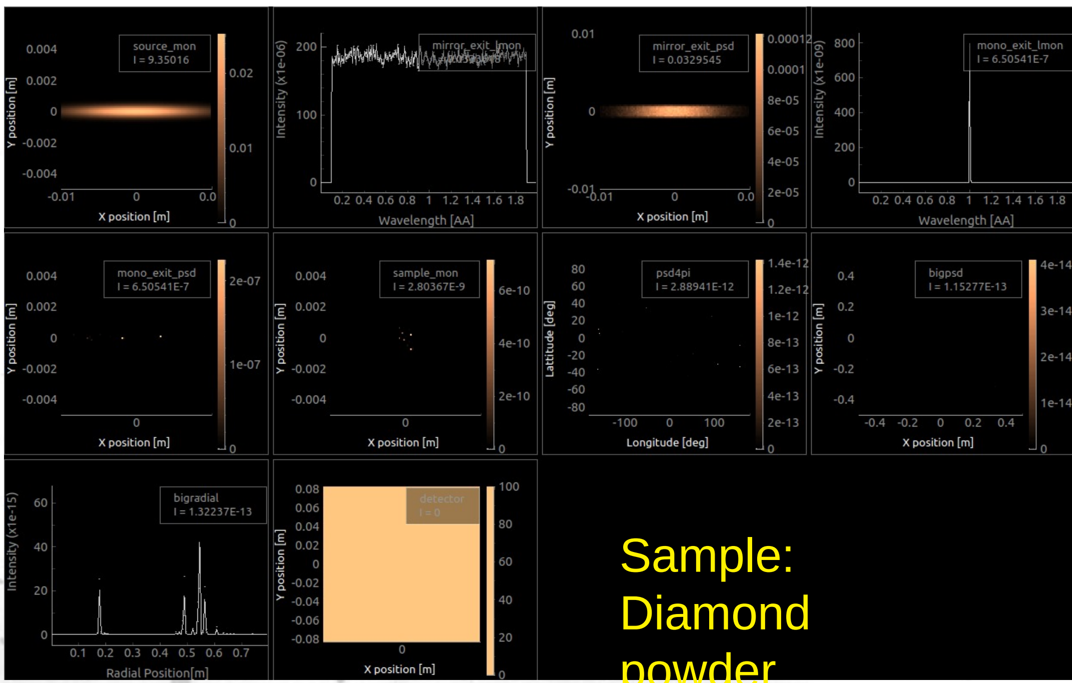
1 s



Max II 711 (MX)

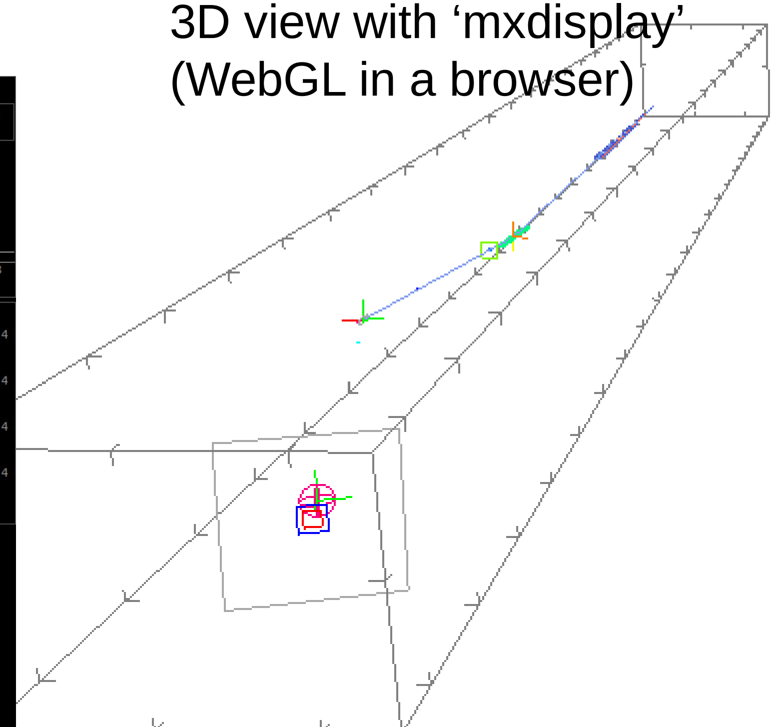
A complete BL in 200 lines.

Execution: **40 s**



Sample:
Diamond
powder

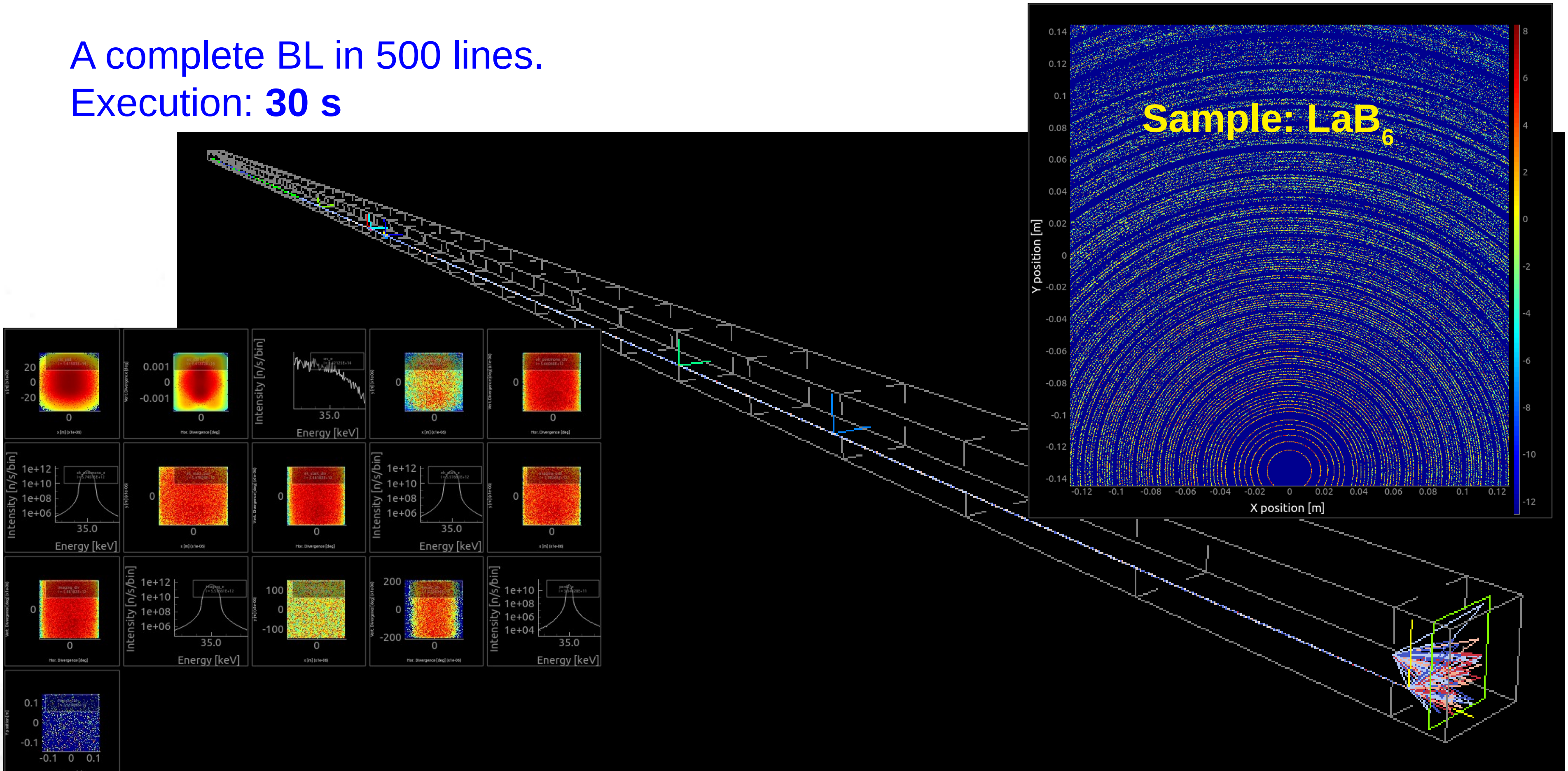
3D view with ‘mxdisplay’
(WebGL in a browser)



Max IV DanMAX (PXRD)

Try it on the web:
https://sim.e-neutrons.org/instrument-menu/xrd/MAXIV_DanMAX_pxrd2d

A complete BL in 500 lines.
Execution: **30 s**



Limitations

Initially oriented towards ‘science’, not machine physics. That’s why we favour interfaces with other packages.

Can handle 100% coherent OR 100% incoherent beam.

BUT

Made to ease contribution and extend. Easy to learn.

Possibility to add new optics, samples and detectors. This is Open Source.

Particle (photon) description can be extended easily (add more state parameters to e.g. better describe coherent beams) to handle *e.g.* polarised BL.

McXtrace

The Future of McXtrace

Contribute to new sample kernels:

- Single Xtal (validate)
- EXAFS
- IXS/XPS/ARPES [*S(**q**,w)* exists]

Contribute to new detector models:

- From MLF-SOM (= MOS-FLM^{-1})
- Interfaces with e.g. HORUS, GEANT4, ...

Add more optics.
Tomography, ptychography...

On-going work to use GPU's *via* NVIDIA OpenACC.

Add interface to OpenPMD (HDF5) particle/ray records
Couple to e.g. OASYS, SIMEX, COMSYL, SRW, … SpotX, Solemio ?

Tutorial/School as a Training for Soleil Staff planned at Soleil: **2-6 December 2019**

How to contribute ?

Using McXtrace is already a good start. You can get a feeling in a few minutes...

You may then build new beam line descriptions. Generally 100-200 lines.

You further may adapt or build new components, *e.g.* 100-200 lines.

Send us an email or push on GitHub to allow others to benefit from your work:

emmanuel.farhi@synchrotron-soleil.fr
erkn@fysik.dtu.dk

http://www.mcxtrace.org/