3Codes Workshop: McXtrace Hands-on Tutorial

June 5, 2013

1 A CRL-stack (aka. Transfocator)

In this exercise we get to know how McXtrace operates, and build a beamline based on compound refractive lenses (CRLs) and examine the focusing properties hereof. A version of the fully equipped beamline may be found in: ex1_CRLs.instr in the 3codes workshop ftp-area.

1.1 Getting started

1. Start the McXtrace GUI by double clicking on the MxGUI icon. You should now see the main window shown below (or similar)

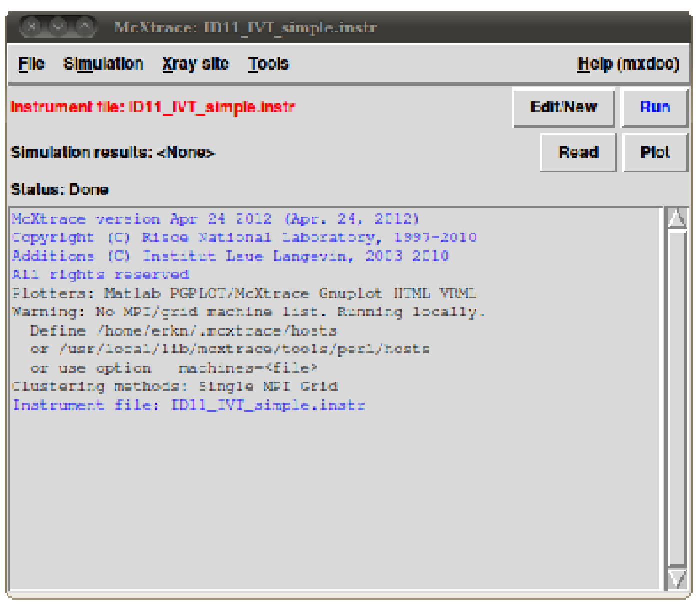

Figure 1: The Main McXtrace GUI window

2. Open a new instrument file using the file menu, then insert a template beamline by using the `Insert/Instrument Template` menu item. Save it in a convenient place through `File/Save`.

3. Insert a source. Position your cursor in the TRACE section below the Progress_bar component and `Insert/Sources/Source_gaussian`. This opens a window similar to figure 2 below:

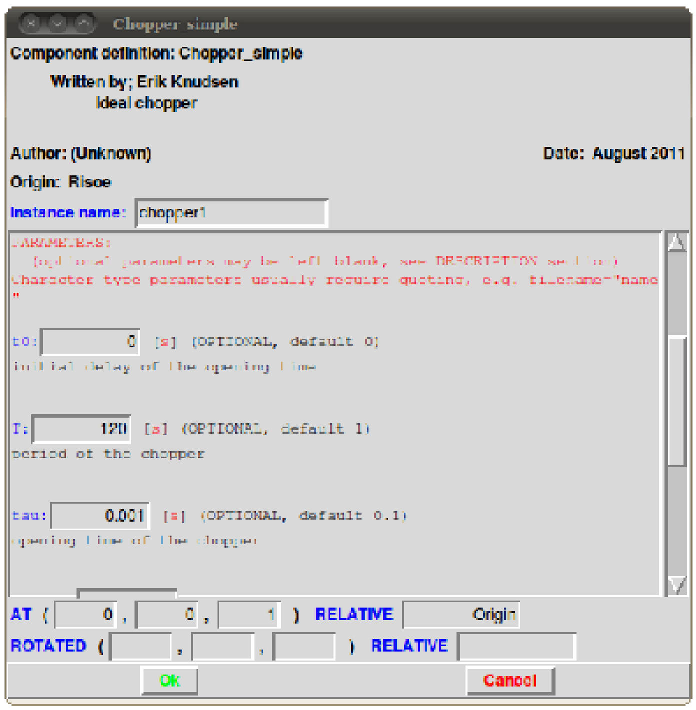

Figure 2: Device/Component insertion dialog window.

Fill in the desired parameters:

sig_x	sig_y	sigPr_x	sigPr_y
48.2e-6 m	9.5e-6 m	100e-6 rad	4.3e-6 rad

gauss	dist	E0	dE
1	31.5 m	23.32 keV	1 keV

,

the coordinates for the source model: (0,0,0), and which object it should be placed in relation to: Origin. Once you're finished press OK. The GUI has now inserted the necessary code for the source model at the cursor. Note that we could just as easily have written this using a text-editor of choice (Emacs, gedit etc...) - McXtrace merely interprets the instrument file.

4. Try out the online component documentation by selecting Help/Component Library index. This should open a browser window with a listing of all available McXtrace components. From this web page you may access the online documentation for components and also the source code for the components should you wish to do so. Try to find the `Source_gaussian` model and what the parameters mean.

1.2 Monitor the inserted source model

1. Insert a 2D-monitor, and an energy spectrum monitor at the CRL-stack position (31.5 m downstream from the source). In our example this translates to a position of (0,0,10) in McXtrace coordinates. Use the components `PSD_monitor` and `E_monitor` to accomplish this. Contrary to other McXtrace components, a Monitor may reside on top of another component, if the `restore_xray` parameter is nonzero.

   Run a simulation to get the beam footprint and spectrum at the CRL-stack position, by pressing the "Run"-button. This should now open a new window with boxes for the defined input parameters of the beamline file:

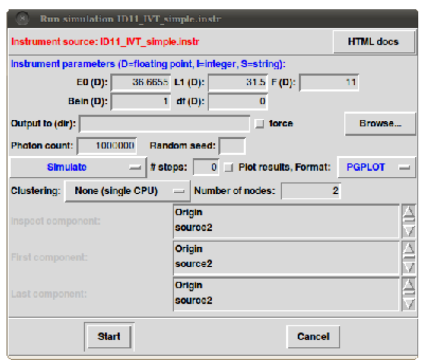

Figure 3: The McXtrace run-window

   To actually run the simulation just press Start. One may name an input directory in the "Output" box, if not McXtrace will generate a subdirectory based on the name of the beamline/instrument file and the current date and time. Once the simulation has run, you should be returned to the main window. To take a look at your newly generated data please press the "Plot" button. If you happen to have multiple CPUs and MPI installed on your machine you may take advantage of this using the "Clustering" drop-down box and choose "MPI".

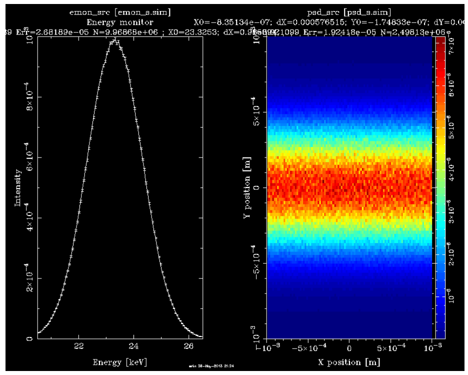

Figure 4: Example of overview plot

Is the result what you expected?

2. Optimize your simulation using directional sampling. Add the parameters *focus_xw*=0.001, *focus_yh*=0.001, and *dist*=31.5 to your source parameters. This will cause McXtrace to only sample that part of the photon phase space which lies on trajectories that passing through a *focus_xw* × *focus_yh* m^2 aperture dist m downstream from the source. The weights of the photons are reduced to avoid biasing - this way we get correct intensity measures inside the sampling window.

1.3 Insert a CRL-stack.

The first stack should be a set of 16 parabolic Be lenses to set the focus at 10 m downstream of the source, the second stack a set of 16 Al lenses. Use `Lens_parab` to set a Be CRL-stack 31.5 m downstream from the source, with the following parameters:

Material	radius of curvature (at tip)	Number of lenses
Be	200 µm	16

Diameter	Thickness (at tip)	Separation between lenses
1 mm	50 µm	2 mm

1.4 Verify that the focus is where you expect

1. Insert a 2D monitor at a variable position downstream of the source to verify that the focus is where you think it is. Make the z-position of the monitor a variable defined as an input parameter in the DEFINE INSTRUMENT line of your instrument file. This way you an avoid recompilation of your simulation every time you change the distance. Furthermore it allows you to perform pointwise scans of the parameter. In the appropriate

boxes in the run-window (see below) you may input a range as min,max and a number of scan points.

2. Use the lens stack together with a slit (`Slit`) as a low resolution monochromator, exploiting the achromatic focusing of the CRL. How would you set the slit? How would you verify the monochromatizing behaviour?

1.5 Visually trace photons through the beamline.

Use mxdisplay to try and visually track photons through the beamline model. In the GUI this is done by selecting “Trace” instead of simulate in the run window. You will need to zoom in a lot to see the CRLs clearly. You should be able to saee something like figure 5 below.

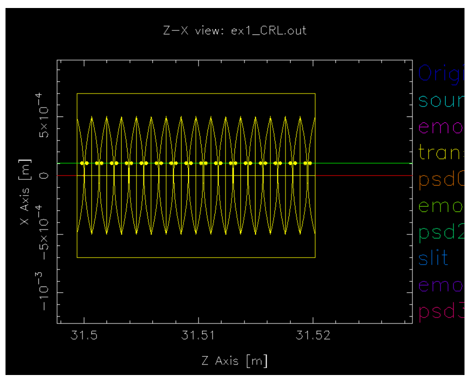

Figure 5: Visual ray tracing through a CRL stack.

1.6 Insert the Al stack of lenses

1. Insert a second stack of CRLs with parameters at the same position as before:

Material	radius of curvature (at tip)	Number of lenses
Al	200 µm	10

Diameter	Thickness (at tip)	Separation between lenses
1 mm	20 µm	1 mm

Both stacks may be included in the instrument file simultaneously, but only one may be active. This may be achieved by adding the WHEN keyword in front of the AT clause in the CRL-definitions. The syntax is: WHEN (c-expression) AT ...

2. How would you make McXtrace dynamically switch between CRL-stacks based on an input variable?

2 Coherence/Slit interference

We shall now turn our attention to tracking the phase of a wavefront decontructed into rays. What we will do is trace rays, emitted from a fully coherent point source through first a single slit, then a double slit arrangement.

2.1 Single slit diffraction

A version of this instrument file is in ex2_Single_slit.instr

1. Set up a point source and a coherent detector 2 m downstream, using the components `Source_pt` and `PSD_monitor_coh`. Note that that if *ny*=1 `PSD_monitor_coh` becomes a 1D-monitor and only displays a curve. The wavelength emitted should be tightly concentrated around 2 Å.
2. Put a slit, *xwidth*=5 µm 1 m downstream from the source. Set the sampling to at least fill your detector. This is done in much the same manner as fo sources using *dist* and *focus_xw*, *focus_yh* parameters.
3. What do you see? Is this what you expect?

2.2 Double slit diffraction

A version of this instrument file is in ex2_Double_slit.instr

1. Exchange the single slit with two 1 µm wide slits with an separation of 4 µm. Note that the sampling window by default is placed directly downstream from the slit. It may be placed at an arbitrary point in space using the parameters *focus_x0* and *focus_y0.*
2. Parallel and mutually exclusive devices are known as GROUPs in McXtrace. Put a statement like:

   `GROUP` <groupname>

   after the `AT` statement of both slits. This tells McXtrace that these components are logically nest to each other and that one excludes the other.
3. How does the grouping affect the outcome? What happens if you don't use a GROUP?
4. Looking back to section 1.6 - can you think of a GROUP-based way of switching between lens-stacks?

2.3 Coherent propagation through CRLs

Start with the example instrument NSLS2_CHX, and modify the monitors to also take the phase of the beam into account. Are the statistics of the results sufficient? If not, increase the statistics, if possible through parallelization. Try to modify the setup to get 1D vertical and horizontal cuts on the coherent monitors. An example is given in ex2_NSLS_CHX_beamline.instr in the workshop mcxtrace tutorial file bundle.

3 Kirkpatrick-Baez focusing

See ex3_KB.instr for a solution

The aim of this exercise is to build a Kirkpatrick Baez focusing system in McXtrace. We will use a simple cylindrical curving mirror model, `Mirror_curved`, to focus the beam in the same manner as exercise 1, i.e. focus the beam at a pt 10 m downstream from the focusing system at 40 m downstream from the source. The best cylindrical curvature that approximates an ellipse is:$\rho = \frac{2f}{sin(\theta)}$, where $1/f = 1/l_1 + 1/l_2$. Assume a deflection angle of 9.7 degrees. This yields a radisu of curvature of approx. 100 m.

`Mirror_curved` is by default set in the XY-plane. Think of how you can set your mirror to deflect the beam in the desired angle using ROTATE. Remember that for a mirror rotated θdegrees, the beam is redirected 2θ. Set up an Arm that points from the mirror downstream.

Verify that your focus plane is where it should be, for instance by scanning a monitor along the optical axis of the beam.

Now set up a 2nd mirror 2 m downstream from the first to focus in the other direction. How can you position this Mirror?

4 Double Crystal Monochromator

A solution is in ex4_DCM.instr

The general idea is inspired by the setup planned for the european XFEL FXE beamline in Hamburg, in that we wish to simulate a pseudo channel-cut double bounce crystal monochromator. A diagram of one design option with the DCM centre of rotation in the middle of the first crystal surface is shown below

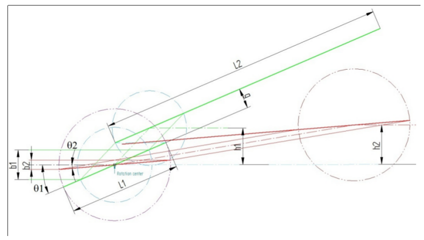

Figure 6: Schematic of DCM option with COR in the middle of 1st crystal.

1. Set up a gaussian source that radiates 12.5 keV radiation.
2. Measure the beam footprint at the Monochromator site, z=100 m. If an Si 111 monochromator is to reflect 12.5 keV, how should you position the first monochromator crystal?
3. How should the 2nd monochromator crystal be set to transmit as much as possible of the beam? And how would you do this in McXtrace? Does the 2nd crystal cut the beam?
4. Measure the exit spectrum.
5. At FXE - a second DCM is envisioned after the Si 111, either a Si 311 or a Si 511. Set up a similar counter-reflecting DCM downstream in your McXtrace instrument and verify its functioning.

5 Time of Flight simulation

Solution example: ex5_TOF.instr

We will in this example use a disordered sample model in conjunction with a simple Chopper model to simulate a generic "Pump-and-Probe" experiment.

1. Start by setting up a source (perhaps by copying work work from an earlier exercise), that emits photons at around 7.5 keV.
2. Add a chopper (`Chopper_simple`) to generate a pulsed signal. The chopper should initially generate an x-ray pulse of 100 ps duration. Presently it is a limitation of the chopper model that only a single pulse can be generated. If jitter and sample variations may be neglected this approximation is valid. Please make sure that the parameter *isfirst* is set to 1.
3. At some point downstream insert a `Molecule_2state` sample using parameters like this:

```
COMPONENT sample=Molecule(
state_0_file="Fe_bpy_GS_DFT.txt",state_1_file="Fe_bpy_ES_DFT.txt",radius=0.01,
psimin=5*DEG2RAD,psimax=45*DEG2RAD, etamin=-1*DEG2RAD,etamax=1*DEG2RAD,
t_relax=100e-9, delta_t=2e-12, excitation_yield=1.0 )
```

The data files referred to, "Fe_bpy_GS_DFT.txt" and "Fe_bpy_ES_DFT.txt“, are included as examples in the McXtrace distribution and so the system wil find them automatically.

4. Add a downstream banana-shaped detector to catch the scattered beam. Suggestion: use a `Monitor_nD` and the keyword “banana” in the option string, or a 1D PSD.

5. Simulate scattered signals while varying the time-delay, the *delta_t*-parameter in the sample molecule, between the excitation time and arrival time of the probing x-ray pulse. Can you see a difference between scattering from excited and relaxed sample? Note that the difference may be very small. What would happen if the x-ray pulse was too long?

6 SHADOW-McXtrace interoperation

Starting with a shadow simulation of a wiggler. Use the McXtrace to inject that into the McXtrace simulation from section1.6.

1. Do you get the expected results?
2. How would you improve statistics?

7 Write your own component

Get coffee, lean back and watch the tutorial :-)