

McXtrace introductory exercise.

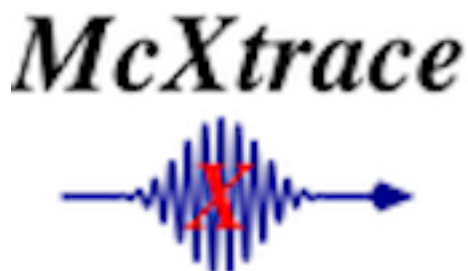


Figure 1: McXtrace

In this exercise we shall build an extremely simple beam-line consisting only of a source, a slit, and a detector/monitor.

Step 1. Open McXtrace.

Start by either typing `mxgui` in a terminal, double clicking the McXtrace icon, or choosing McXtrace from the application/start menu. The McXtrace main window should now open, looking like this.

Step 2. Start a new instrument description.

In the 'File' menu, please choose the 'New Instrument' item. This opens a file dialog window where you can choose a name for your new instrument file (e.g. `test.instr`). Do so and click *Save*.

A McXtrace instrument description is a text-file just like any other program you write. Generally the suffix `.instr` is used but you could use whatever you like for this.

This opens the Editor window of the GUI where you can edit a very simple template instrument file. You may equally open the model with any text editor.

:warning: There is a **bug** with the 3.5.1 `mxgui`. The initial opening of the new file may break and exit `mxgui`. Restart `mxgui` and reopen the new file. An external editor window will pop-up (e.g. `gedit`). The ability to insert pre-defined components is not available.

Step 3. Insert a source.

The editor can help you write instrument files, but really is just an editor. If you prefer to use something else, this works equally fine.

Some of the allowed McXtrace code blocks have already been defined in the template file. You can identify the usual sections and keywords:

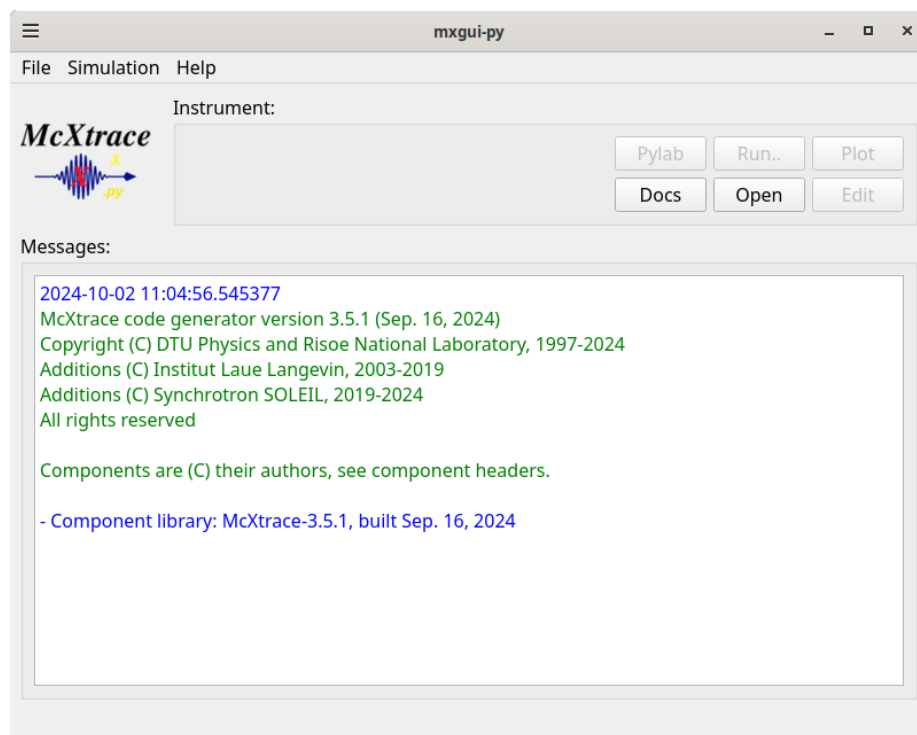


Figure 2: McXtrace main window

```

1 /*****
2 *. Instrument: <instrument.name>
3 *
4 *. %Identification
5 *. Written by: <your.name (email)>
6 *. Date: <current.date>
7 *. Origin: <your.institution>
8 *. %INSTRUMENT_SITE: Templates
9 *
10 *. [Put here the instrument short description]
11 *
12 *. %Description
13 *. [Put here a longer instrument description]
14 *
15 *. Example: <parameters=values>
16 *
17 *. %Parameters
18 *. parameter1: [unit].parameter1.description
19 *. ...
20 *
21 *. %L
22 *. <put here any reference/HTML link>
23 *
24 *. %End
25 *****/
26 DEFINE INSTRUMENT.template_simple(parameter1=1)
27
28 DECLARE
29 %{
30 %}
31
32 USERVARS
33 %{
34 %}
35
36 INITIALIZE
37 % f

```

C Largeur des tabulations : 2 Lig 1, Col 1 INS

Figure 3: McXtrace editor window

- a header that describes the purpose of the model. Please keep information up-to date, especially the input parameter details.
- the **DEFINE INSTRUMENT** line that gives the name of the model, and its input parameters.
- the **DECLARE** section where C-style variables may be listed, and may be used elsewhere.
- the **USERVARS** section where to define C-style variables attached to each particle.
- the **INITIALIZE** section which contains C-code to execute before shooting the first particle.
- the **TRACE** section which contains a ordered list of **COMPONENT** lines.
- the **FINALLY** section which contains C-code to execute at the end.
- the **END** keyword.

Only the **DEFINE INSTRUMENT**, **TRACE** and **END** parts are mandatory.

Position your cursor below the keyword **TRACE** and insert the code:

```
COMPONENT src = Source_pt()
AT (0,0,0) ABSOLUTE
```

This way we have positioned a point source called **src** (this name is fully yours), for which some parameters must be defined. Not all parameters are required, and often there are different ways to specify the same information. Component parameters are usually a set of:

- geometrical (width, height, shape, ...)
- physical (energy, density, ...)
- computational (probabilities, statistical focusing)
- input/output (file names, ...)

It is recommended to open the documentation by clicking the ‘Docs’ button in the main interface. A browser window will show-up.

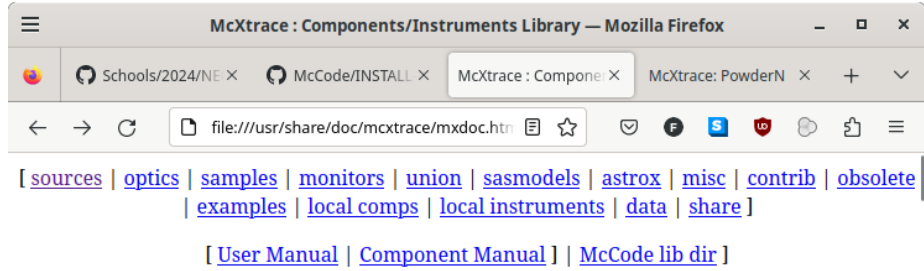
Step 4. Fill in some source parameters.

It is first required to get documentation about the components to be inserted.

Most components contain an **Example:** line which provides a basic syntax with main parameters.

Select the component **Source_pt** from the component list (browser). This is a point source, which illumination is isotropic. Looking at the **Source_pt** documentation, we select:

- focus_xw=1e-4
- focus_yh=1e-4
- dist=2
- E0=15
- dE=1

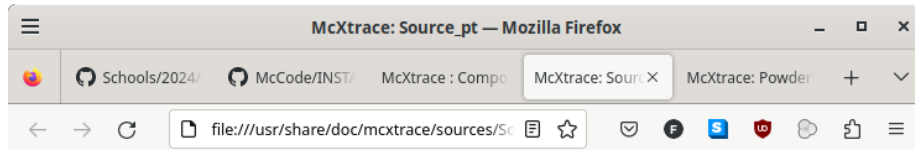


Components and Instruments from the Library for *McXtrace*

Sources

Name	Origin	Author(s)	Source code	Description
Bending magnet	DTU Physics	Erik B. Knudsen	comp	Model of a bending magnet source
Source_div	Risoe	Erik Knudsen	comp	X-ray source with Gaussian or uniform divergence
Source_div_quasi	DTU Physics	Mads Carlsen and Erik Bergbäck Knudsen (erkn@fysik.dtu.dk)	comp	Quasi-stochastic X-ray source with Gaussian or uniform divergence
Source_flat	Risoe	Erik Knudsen	comp	A flat rectangular or circular surface emitting x-rays
Source_gaussian	NBI	Jana Baltser & Erik Knudsen	comp	Gaussian cross-section source
Source_genesis13	Risoe	Erik B Knudsen	comp	Interface source for importing GENESIS 1.3 generated X-ray

Figure 4: mxdoc Component Reference



[[Identification](#) | [Description](#) | [Input parameters](#) | [Links](#)]

The source_pt Component

An x-ray point source

Identification

- **Site:**
- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** June 29th, 2009

Description

A simple source model emitting photons from a point source uniformly into 4pi. A square target centered on the Z-axis restricts the beam to that aperture.
 If an input spectrum datafile (spectrum.file) is not specified, the beam is restricted to emit photons
 The input spectrum file should be formatted such that x-ray energy/wavelength is in the first column and
 lines starting with # are considered part of the file header. If a datafile is given, a nonzero E=0 va
 as opposed to wavelength (in AA). Wavelength is the default.
 Flux is given in the unit photons/s

Example: Source_pt(dist=1,focus_xw=0.1,focus_yh=0.1, lamda=0.231, dlamba=0.002)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default	CompInstanceName
focus_xw	m	Width of target	0	

Figure 5: mxdoc component view

so that the final component description should be:

```
COMPONENT src = Source_pt(focus_xw=1e-4, focus_yh=1e-4, dist=2, E0=15, dE=1)
AT (0,0,0) ABSOLUTE
```

These settings mean that the beam size will be 1x1 mm² at 2 m distance (along Z), and the beam energy will be centred around 15 keV.

Step 5. Insert a Slit and a Monitor.

Now that we have generated some photons, we shall shape the beam, and measure the spatial distribution.

With the same methodology, insert a `Slit` component 0.7x0.7 mm² at 2 m, and then a `PSD_monitor` component at 5 m. Set the monitor width and height so that the beam spot spreads on most of the surface.

We now have a complete (albeit too simple) model which we can run. Save the file and click on the `Run`-button in the main GUI.

You should now be presented with a new dialog asking you to set some parameters for your simulation. For now the default should suffice, so click the `Run` button to start the simulation. If all goes well, this should return us to the main window where we may click `Plot` once the simulation is finished. This ought to give you a plot like:

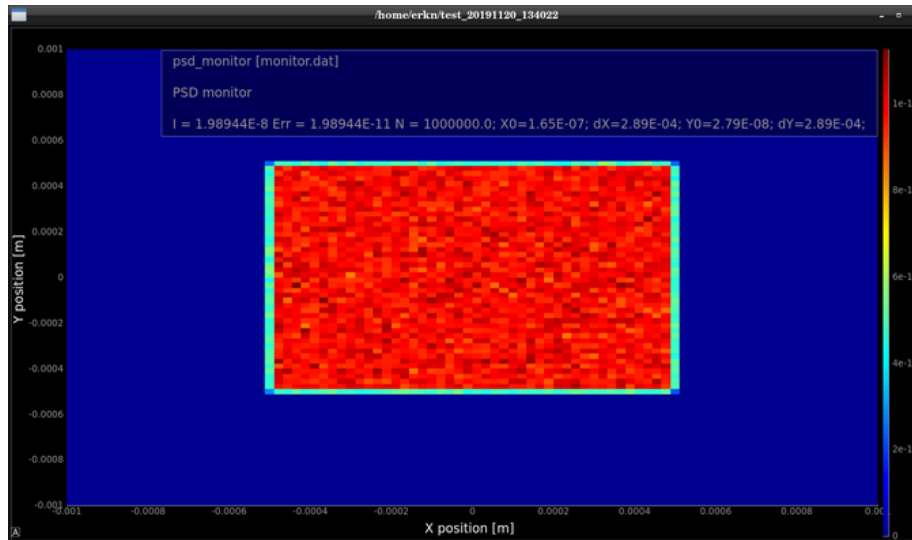


Figure 6: First Simulation Results

Step 6. Add more monitors and parameterise the model.

It is possible to change the monitor type (and its parameters), or add others.

We suggest that you add an `DivE_monitor`, i.e. a monitor which first axis is Energy and 2nd is the horizontal divergence. Position this new component a few cm away after the previous one, and set some reasonable parameters.

Question: how would you get an energy/vertical divergence monitor ? (tip: ROTATED).

Now, we wish to be able to change the source energy without changing the model description file. The `DEFINE INSTRUMENT template_simple(parameter1=1)` is there for that purpose. Change the default `parameter1=1` (which is not used here), by e.g.:

```
E0=15
```

This syntax sets an `E0` parameter with a default value of 15. You may at the same time change the name of the model to match your file name (e.g. `test.instr` -> “`DEFINE INSTRUMENT test`”).

Now use the `E0` model parameter in the `Source_pt` parameters. It could even be used for the `DivE_monitor(Emin=..., Emax=...)`, so that the monitor adapts its range as a function of the source.

Save and run the model again, and look at the results by clicking the Plot button. As you can now see, the input model parameter is now `E0`, set to 15 at start, but allowing to change its value without touching the model description.

The two generated files from the monitors are text based.