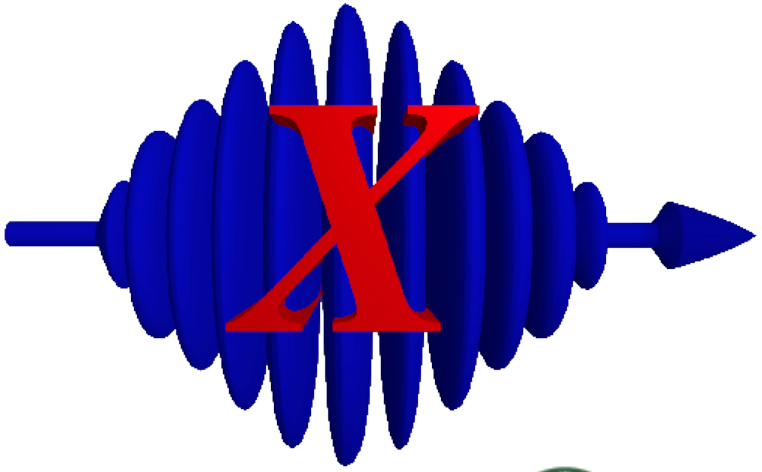
McXtrace



FAQ, advanced grammar, tips and tricks

Peter Willendrup (pkwi@fysik.dtu.dk)

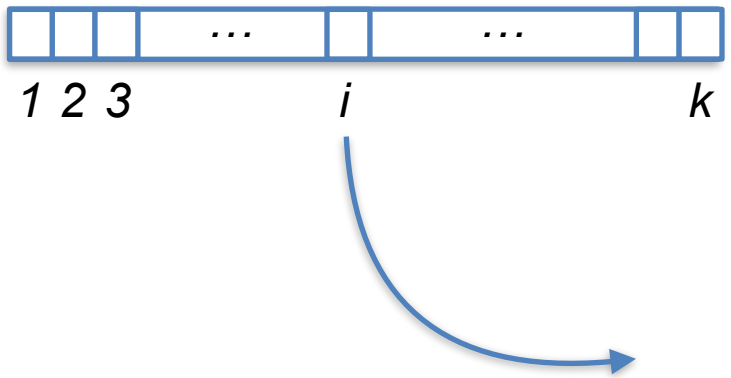
What is really the information content…?

- McXtrace sources generally provide "intensity" in units of photons/s (into a chosen solid angle)

- That intensity is carried through the instrument on a discrete set of "X-ray rays"

In a histogram sense

- Imagine a histogram, e.g. I(λ)



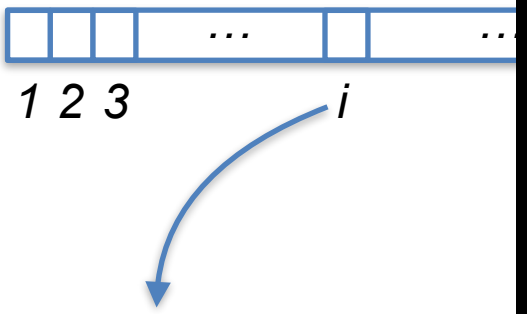
In bin i, N events each carrying a fractional intensity p_j so that

$$I = \sum_N p_j$$

- The RMS variance over that set becomes our statistical error bar *E*

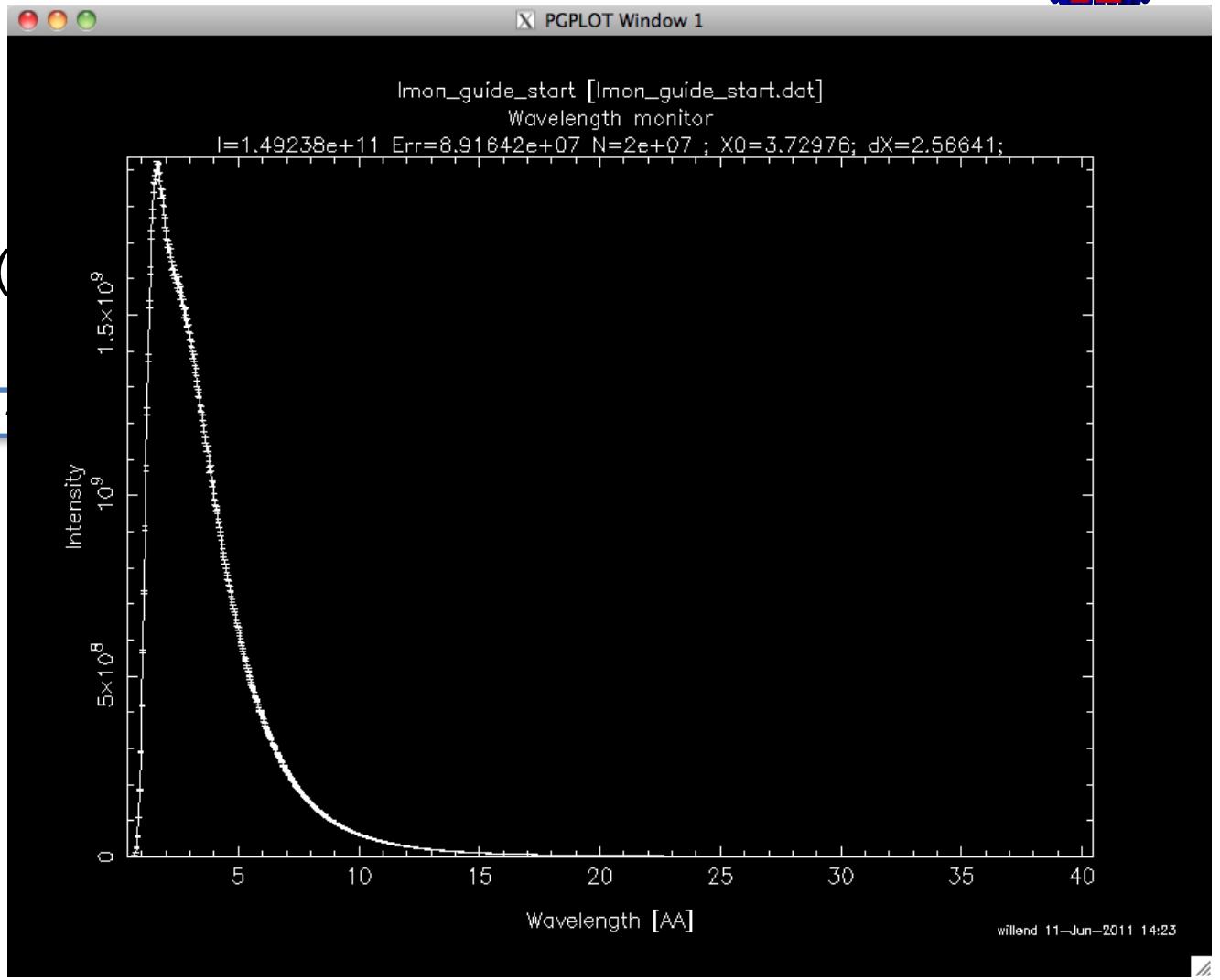
In a histogram sense

Imagine a histogram, e.g. I(



In bin i, N events each carrying a fractional intensity p_j so that

$$I = \sum_N p_j$$



The RMS variance over that set becomes our statistical error bar *E*

From "Virtual experiments - the ultimate aim of neutron ray-tracing simulations", K. Lefmann et al., Journal of Neutron Research 16, 97-111 (2008)

Let n be the number of neutron rays reaching the detector, and let the rays have (different) weights, w_i. The simulated intensity is then given by

$$I = \sum_{i=1}^{n} w_i. \tag{1}$$

The estimate of the error on this number is calculated in the McStas manual [1], and the standard deviation is approximated by

$$\sigma^2(I) = \sum_{i=1}^{n} w_i^2. \tag{2}$$

In real experiments, $w_i = 1$, whence we reach $I = n$ and $\sigma(I) = \sqrt{I}$ as expected (for counts exceeding 10). Let the virtual time be denoted by t. The simulated counts during this time becomes

$$C = tI, \tag{3}$$

From "Virtual experiments - the ultimate aim of neutron ray-tracing simulations", K. Lefmann et al., Journal of Neutron Research 16, 97-111 (2008)

and its error bar estimate is

$$\sigma^2(C) = t^2\sigma^2(I). \tag{4}$$

However, to simulate a realistic counting statistics, we must fulfill

$$\sigma_{\mathrm{VE}}(C_{\mathrm{VE}}) = \sqrt{C_{\mathrm{VE}}}. \tag{5}$$

This is obtained by adding to (3) a Gaussian noise $E(\Sigma)$ of mean value zero and standard deviation Σ:

$$C_{\mathrm{VE}} = tI + E(\Sigma). \tag{6}$$

The standard deviation for the VE becomes

$$\sigma^2_{\mathrm{VE}}(C) = t^2\sigma^2(I) + \Sigma^2. \tag{7}$$

Now, the requirement (5) allows us to determine Σ:

$$\Sigma^2 = tI - t^2\sigma^2(I). \tag{8}$$

Since Σ^2 must remain positive, we reach an upper limit on t

$$t_{\max} = \frac{I}{\sigma^2(I)}. \tag{9}$$

Sketch of an algorithm…

1. On a given McXtrace histogram
2. For the non-zero bins, calculate $$t_{\max} = \frac{I}{\sigma^2(I)}.$$
3. The *smallest* $t_{\max}$ defines the "maximal counting time" allowed by your statistics
4. Preferably a "background" should be added - use a "known experimental value" or an estimate…

Important points to remember

1. Your simulation will only contain elements you provided / defined

2. … to the precision you defined

3. Answers the questions you posed

4. Background essentially only from “sample”, or sample-near objects

Errorbars → maximum "integration time"

Lefmann, K., Willendrup, P. K., Udby, L., Lebech, B., Mortensen, K., Birk, J. O., ... Farhi, E. (2008). *Virtual experiments: the ultimate aim of neutron ray-tracing simulations.* Journal of Neutron Research, 16(4-4), 97. https://doi.org/10.1080/10238160902819684

Plus the attached note from Annette Vickery:

A note about McStas errorbars

A. Vickery, P. K. Willendrup and E. Knudsen

January 31, 2014

Contents

Onto efficiency…

- Apply focusing techniques
  - At the source (spatially, temporally, in wavelength…)
  - At the sample, if possible
- (carefully!) Apply SPLIT - but only if immediately followed by Monte Carlo choices, e.g. in sample
- Alternatively use MCPL o/i which allows repetition - beware of biases!

Onto efficiency…

- Apply focusing techniques
  - At the source (spatially, temporally, in wavelength…)
  - At the sample, if possible
- (carefully!) Apply SPLIT - but only if immediately followed by Monte Carlo choices, e.g. in sample
- Alternatively use MCPL o/i which allows repetition - beware of biases!

All of this can be considered "variance reduction"

or biasing

Onto efficiency…

- Use MPI parallelisation - included in macOS install, easy to get on Linux…
- The Intel C compiler is known to give ~factor of 2 wrt. gcc in most cases
- - **Still** consider if you are asking the right question if runtimes reach days/weeks…

Cluster utility scripts

- mcsub cluster scripts

```
./mcsub_slurm.pl
Usage: ./mcsub_slurm.pl [options] [mcrun params]
 -h         --help             Show this help
 -rN        --runtime=N        Specify maximum runtime (hours) [default 1]
 -qQNAME    --queue=QNAME      Specify wanted SLURM queue [default 'express']
 -e<mail>   --email=<mail>     Specify address to notify in reg. sim status [default
none]
            --nodes=NUM        Specify wanted number of nodes [default 1]
            --name=NAME        Specify slurm job name [default
"McSub_<USERNAME>_<TIMESTAMP>"]


After running ./mcsub_slurm.pl NAME.batch is ready for submission using the sbatch
command
```

- Takes a “mxrun commandline”

- Writes batch file “template” for use with PBS or slurm cluster queue systems

- https://github.com/McStasMcXtrace/McCode/tree/master/tools/cluster-scripts

Examples Directory

- *Simple grep →*
  - *25 examples of Monitor_nD*
  - *4 examples of Source_lab*
  - *4 examples of Undulator*
  - *5 examples of PowderN*

Unix/Linux:
*grep Monitor_nD /usr/share/mcxtrace/1.5/examples/**

- *Simple grep →*

They have an author name!

For instance:

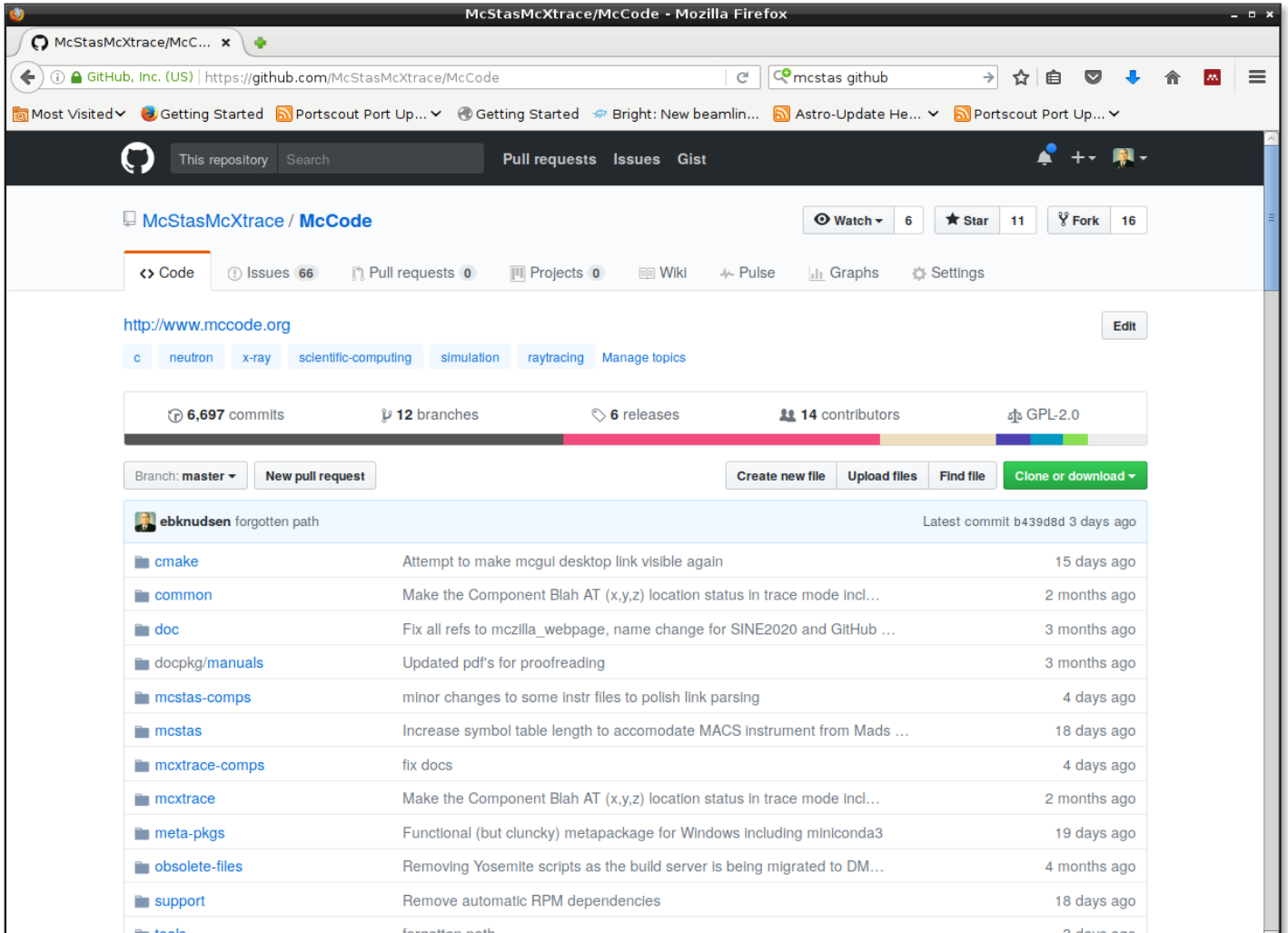
DBD_IBM_Si_analyzer → Marcus H. Mendenhall

Test_SX → E. Farhi

Pump_probe → E. Knudsen

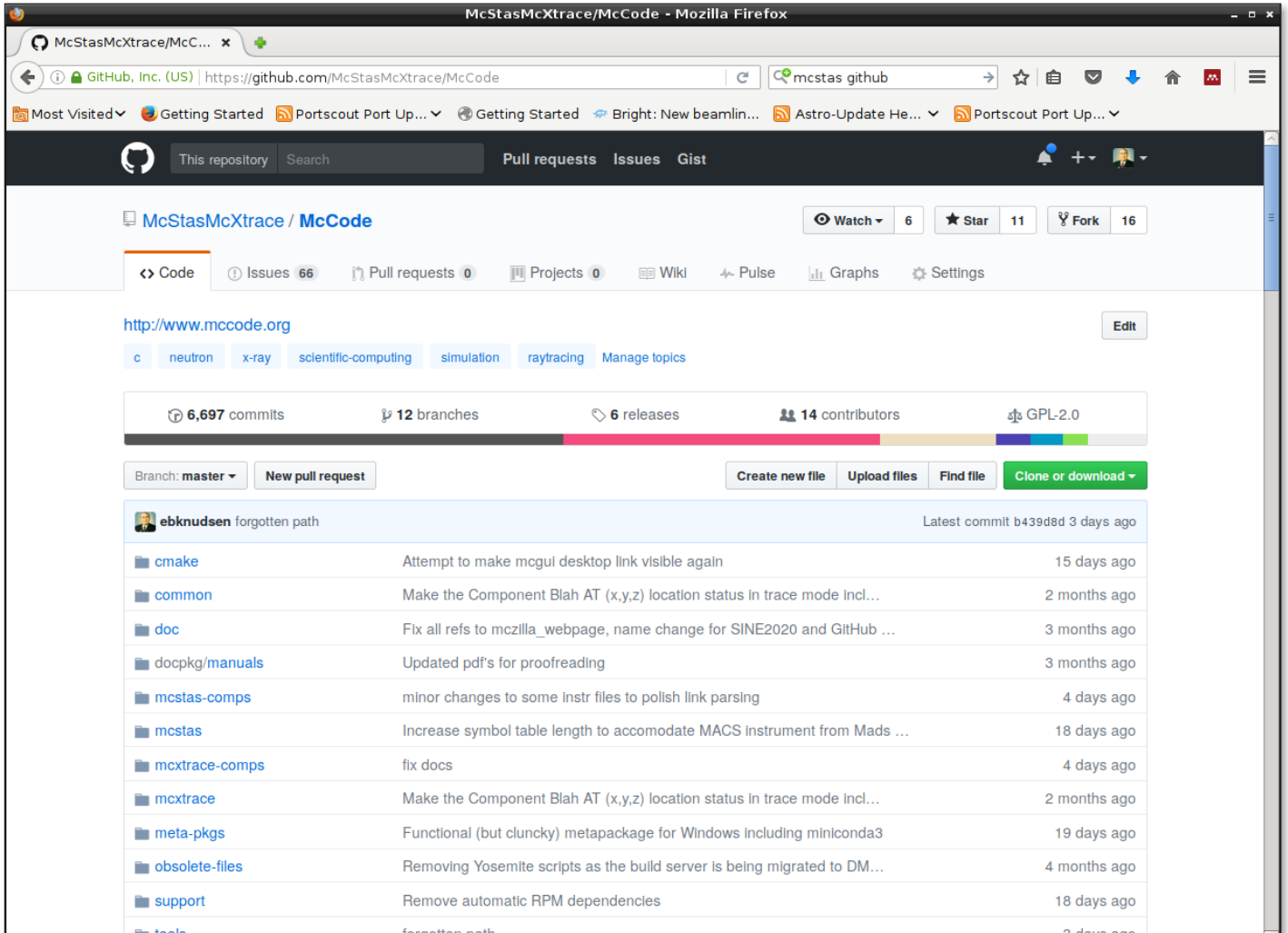
GitHub.com

https://github.com/McStasMcXtrace/McCode



GitHub.com

https://github.com/McStasMcXtrace/McCode



GitHub.com: wiki User- and developer docs

Welcome to the McCode wiki!

General information

- How McStas/McXtrace works overview
- Known issues and FAQ's
- Access to the code tree

Developer guides

- Developer literature
- Building McStas/McXtrace
- Builds and platforms overview
- Release procedure/notes
- UI test checklist
- How to use Eclipse with PyDev
- Debugging the c-code

For users & comp developers

- McStas tutorial: simplified SANS instrument
- Writing Components
- Single crystal - and generating its input
- Generate Vitess modules via mcstas2vitess

Interlinks with other software

- McStas and Mantid
- How to update the MCPL shipped with McStas/McXtrace (McStas and MCNP/Geant4/SIMRES/...)
- MCPL-related issue on Ubuntu 17.04 (McStas and MCNP/Geant4/SIMRES/...)

Documentation on the McCode tools

- User documentation for the 2017- Python tool set'
- mcrun variants - table overview
- mcplot variants - table overview
- mcdisplay variants - table overview

Guides for McWeb (e-neutrons.org simulator and more)

- https://github.com/McStasMcXtrace/McWeb/wiki

Pages 33

Find a Page...

- Home
- Access to the code tree
- Building McStas McXtrace
- Builds and platforms overview
- Debugging the c code
- Developer literature
- Generate Vitess modules via mcstas2vitess
- How McStas McXtrace works overview
- How to update the MCPL shipped with McStas McXtrace
- How to use Eclipse with PyDev
- Known issues and FAQ's
- mcdisplay pyqtgraph
- mcdisplay variants table overview
- mcdisplay webgl
- mcgui

Show 18 more pages...

+ Add a custom sidebar

Clone this wiki locally

https://github.c

GitHub.com: wik

Welcome to the McCode wiki!

General information

- How McStas/McXtrace works overview
- Known issues and FAQ's
- Access to the code tree

Developer guides

- Developer literature
- Building McStas/McXtrace
- Builds and platforms overview
- Release procedure/notes
- UI test checklist
- How to use Eclipse with PyDev
- Debugging the c-code

For users & comp developers

- McStas tutorial: simplified SANS instrument
- Writing Components
- Single crystal - and generating its input
- Generate Vitess modules via mcstas2vitess

Interlinks with other software

- McStas and Mantid
- How to update the MCPL shipped with McStas/McXtrace (McSta MCNP/Geant4/SIMRES/...)
- MCPL-related issue on Ubuntu 17.04 (McStas and MCNP/Geant4/SIMRES/...)

Documentation on the McCode tools

- User documentation for the 2017- Python tool se
- mcrun variants - table overview
- mcplot variants - table overview
- mcdisplay variants - table overview

Guides for McWeb (e-neutrons.org simulator and mo

- https://github.com/McStasMcXtrace/McWeb/wiki

mcdisplay variants table overview

Edit | New Page

Peter Willendrup edited this page on Jan 18, 2018 · 7 revisions

Default trace visualiser "mcdisplay" indicated in bold below

More columns if you scroll... ------>

Variant	Type	2D/3D	Special cmdline switches
mcdisplay-pyqtgraph	Python 3 + pyqtgraph	2D x 3 planes	`--invcanvas` (invert colors)
mcdisplay-webgl	Python 3 + WebGL	Full 3D	`--nobrowse` to avoid spawning broswser
mcdisplay-mantid	Python 3 + Mantid	3D	None
mcdisplay.pl (mcplot-pl on Windows)	Perl 5 + PGPLOT	2D x 3 with PGPLOT, 3D with Matlab	`-pPLOTTER --format=PLOTTER` can be used to forward output to PGPLOT, Gnuplot, Matlab, ... `-ps/-psc/-gif` save hardcopy `--complete` When outputting XML, also describe component geometry `--tmax=VAL` ToF axis limit when in --TOF mode
Plotters	below	this	not

Pages 33

Find a Page...

- Home
- Access to the code tree
- Building McStas McXtrace
- Builds and platforms overview
- Debugging the c code
- Developer literature
- Generate Vitess modules via mcstas2vitess
- How McStas McXtrace works overview
- How to update the MCPL shipped with McStas McXtrace
- How to use Eclipse with PyDev
- Known issues and FAQ's
- mcdisplay pyqtgraph
- mcdisplay variants table overview
- mcdisplay webgl
- mcgui

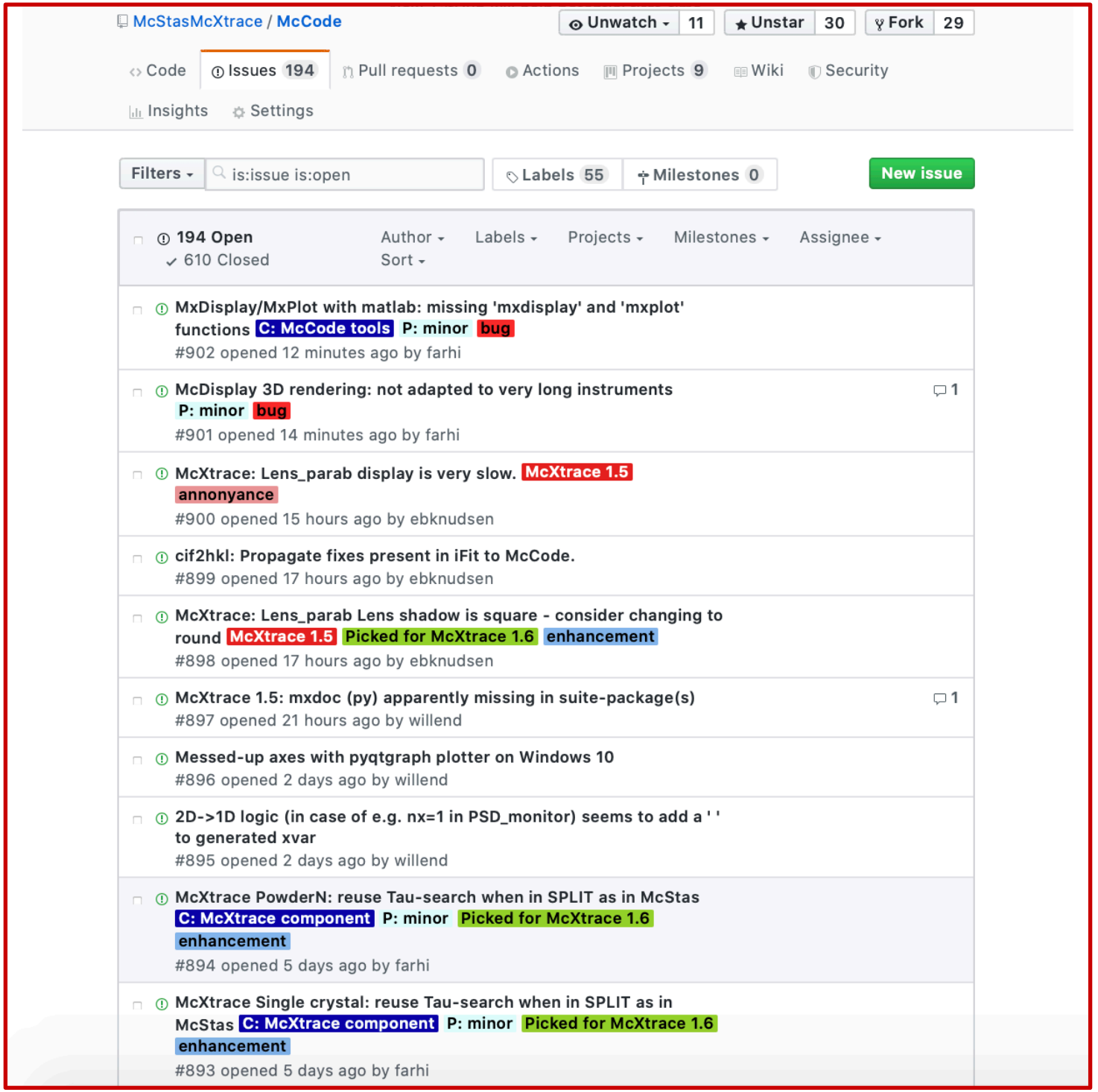
Show 18 more pages...

+ Add a custom sidebar

Clone this wiki locally

https://github.c

GitHub.com: Issue Tracker



McStasMcXtrace / McCode

Unwatch 11 | Unstar 30 | Fork 29

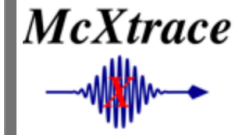
Code | Issues 194 | Pull requests 0 | Actions | Projects 9 | Wiki | Security | Insights | Settings

Filters | is:issue is:open | Labels 55 | Milestones 0 | New issue

194 Open | 610 Closed | Author | Labels | Projects | Milestones | Assignee | Sort

- MxDisplay/MxPlot with matlab: missing 'mxdisplay' and 'mxplot' functions C: McCode tools P: minor bug
  #902 opened 12 minutes ago by farhi
- McDisplay 3D rendering: not adapted to very long instruments P: minor bug 1
  #901 opened 14 minutes ago by farhi
- McXtrace: Lens_parab display is very slow. McXtrace 1.5 annonyance
  #900 opened 15 hours ago by ebknudsen
- cif2hkl: Propagate fixes present in iFit to McCode.
  #899 opened 17 hours ago by ebknudsen
- McXtrace: Lens_parab Lens shadow is square - consider changing to round McXtrace 1.5 Picked for McXtrace 1.6 enhancement
  #898 opened 17 hours ago by ebknudsen
- McXtrace 1.5: mxdoc (py) apparently missing in suite-package(s) 1
  #897 opened 21 hours ago by willend
- Messed-up axes with pyqtgraph plotter on Windows 10
  #896 opened 2 days ago by willend
- 2D->1D logic (in case of e.g. nx=1 in PSD_monitor) seems to add a ' ' to generated xvar
  #895 opened 2 days ago by willend
- McXtrace PowderN: reuse Tau-search when in SPLIT as in McStas C: McXtrace component P: minor Picked for McXtrace 1.6 enhancement
  #894 opened 5 days ago by farhi
- McXtrace Single crystal: reuse Tau-search when in SPLIT as in McStas C: McXtrace component P: minor Picked for McXtrace 1.6 enhancement
  #893 opened 5 days ago by farhi

Mailing List



McXtrace - An X-ray ray-trace simulation package

McXtrace

Download
Components
Linux Installation (deb/rpm)
Mac OSX Installation
Unix Install (src code)
Windows Installation

Project Status
Project Partners
Project People
Goal

Mailing list

Links

Publications

Minutes of Meetings

McXtrace Art

About McXtrace

Documentation
Commands
Tutorial

Search

`mcxtrace-users` mailing list

To subscribe to mcxtrace-users, fill in your email adress and press subscribe:

Subscribe

To access the mailing list interface, go to https://mailman2.mcxtrace.org/mailman/listinfo/mcxtrace-users

To access archived messages, go to https://mailman2.mcxtrace.org/pipermail/mcxtrace-users.

To search in the mailing list archive you may uses: added to the website!

Printable version

Last Modified:

Printable version

Last Modified: Monday, 18-Nov-2019 11:05:20 CET

Mailing List

mcxtrace-users --

About mcxtrace-users | English (USA)

To see the collection of prior postings to the list, visit the mcxtrace-users Archives.

Using mcxtrace-users

To post a message to all the list members, send email to mcxtrace-users@mailman2.mcxtrace.org.

You can subscribe to the list, or change your existing subscription, in the sections below.

Subscribing to mcxtrace-users

Subscribe to mcxtrace-users by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. This is a private list, which means that the list of members is not available to non-members.

Your email address:	
Your name (optional):	

You may enter a privacy password below. This provides only mild security, but should prevent others from messing with your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext.

If you choose not to enter a password, one will be automatically generated for you, and it will be sent to you once you've confirmed your subscription. You can always request a mail-back of your password when you edit your personal options. Once a month, your password will be emailed to you as a reminder.

Pick a password:	
Reenter password to confirm:	
Which language do you prefer to display your messages?	English (USA)
Would you like to receive list mail batched in a daily digest?	(•) No ( ) Yes

Subscribe

mcxtrace-users Subscribers

(*The subscribers list is only available to the list members*.)

Enter your address and password to visit the subscribers list:

Address: Password: Visit Subscriber List

Mailing List - Archive

The mcxtrace-users Archives

You can get more information about this list.

Archive	View by:	Downloadable version
Fourth quarter 2019:	[Thread] [Subject] [Author] [Date]	[Text 5 KB]
Fourth quarter 2017:	[Thread] [Subject] [Author] [Date]	[Text 2 KB]
Second quarter 2017:	[Thread] [Subject] [Author] [Date]	[Text 4 KB]
Fourth quarter 2016:	[Thread] [Subject] [Author] [Date]	[Text 9 KB]
Third quarter 2016:	[Thread] [Subject] [Author] [Date]	[Text 30 KB]
First quarter 2016:	[Thread] [Subject] [Author] [Date]	[Text 3 KB]
Fourth quarter 2015:	[Thread] [Subject] [Author] [Date]	[Text 17 KB]
Third quarter 2015:	[Thread] [Subject] [Author] [Date]	[Text 28 KB]
First quarter 2014:	[Thread] [Subject] [Author] [Date]	[Text 1 KB]
Fourth quarter 2013:	[Thread] [Subject] [Author] [Date]	[Text 2 KB]
First quarter 2013:	[Thread] [Subject] [Author] [Date]	[Text 4 KB]
Fourth quarter 2012:	[Thread] [Subject] [Author] [Date]	[Text 4 KB]
Third quarter 2012:	[Thread] [Subject] [Author] [Date]	[Text 6 KB]
Fourth quarter 2011:	[Thread] [Subject] [Author] [Date]	[Text 2 KB]
Fourth quarter 2010:	[Thread] [Subject] [Author] [Date]	[Text 1 KB]
Third quarter 2009:	[Thread] [Subject] [Author] [Date]	[Text 969 bytes]
First quarter 2009:	[Thread] [Subject] [Author] [Date]	[Text 3 KB]

User forum and help since 2009!!

Spirit of the mailinglist:

- Very friendly and open!
- There is **no such thing** as a stupid question!
- Often answers are given from users rather than developers, especially in areas that include very specific domain science.
- contact: mcxtrace-users@mcxtrace.org

SPLIT, EXTEND, WHEN, COPY & GROUP

(… and the dangerous JUMP)

- Advanced language features:

Tips and tricks for your instrument

Syntax in one, complex view...

McXtrace

SOLEIL SYNCHROTRON

{SPLIT} COMPONENT name = comp(parameters) {WHEN condition}
AT (...) [RELATIVE [reference|PREVIOUS] | ABSOLUTE]
{ROTATED {RELATIVE [reference|PREVIOUS] | ABSOLUTE} }
{GROUP group_name}
{EXTEND C_code}
{JUMP [reference|PREVIOUS|MYSELF|NEXT] [ITERATE number_of_times
| WHEN condition] }

DECLARE / INITIALIZE

- Use the DECLARE section define user variables and functions.

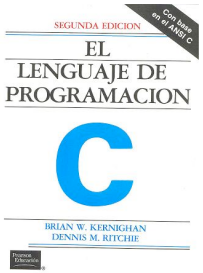
```
DECLARE %{
  double myvar;
%}
```

- Use INITIALIZE for initialization of user variables and calculations.

```
INITIALIZE %{
  myvar = sqrt(PI*input_var)*rand01();
%}
```

- - Both use normal c-syntax.

K & R. / GNU



- BEWARE: (example) What you do in the c-style areas is c-standard, e.g. trigonometric functions from math.h use radians! - McXtrace placement specifiers work in degrees, etc...

DECLARE / INITIALIZE

Useful physics constants:

```
#define CELE      1.602176487e-19   /* [C] Elementary charge CODATA 2006*/
#define M_C       299792458         /* [m/s] speed of light CODATA 2006*/
#define E2K       0.506773091264796 /* Convert k[1/AA] to E [keV] (CELE/(HBAR*M_C)*1e-10)*1e3 */
#define K2E       1.97326972808327  /*Convert E[keV] to k[1/AA] (1e10*M_C*HBAR/CELE)/1e3 */
#define RE        2.8179402894e-5   /*[AA] Thomson scattering length*/
```

plus e.g. DEG2RAD, RAD2DEG, and these math constants

```
(#  define PI 3.14159265358979323846 )
 #  define M_PI PI
 #  define M_PI_2 M_PI/2.0
 #  define M_PI_4 M_PI/4.0
 #  define M_1_PI 1.0/M_PI
 #  define M_2_PI 2*M_1_PI
 #  define M_2_SQRTPI 2/sqrt(M_PI)
 #  define M_SQRT2 sqrt(2)
 #  define M_SQRT1_2 sqrt(1/2)
 # endif
```

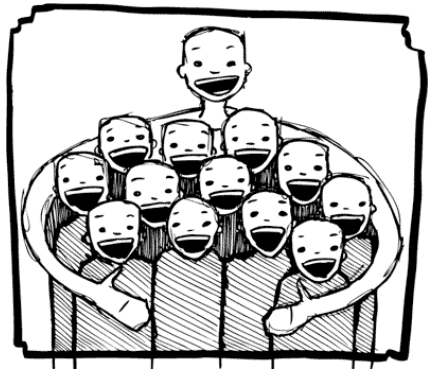
plus anything you can imagine in terms of trigonometric functions from C <math.h>
(beware, these take radians as input - as opposed to our ROTATED statements)

%include

- Instrumentfiles can include external c-code or other instrumentfiles... (These are examples from McStas)

- ILL_H15_IN6.instr:%include "monitor_nd-lib"
- ILL_H16_IN5.instr:%include "ILL_H16.instr"
- ILL_H25_IN22.instr:%include "ILL_H25.instr"
- ILL_H25_IN22.instr:%include "templateTAS.instr"

- Used in the DECLARE section



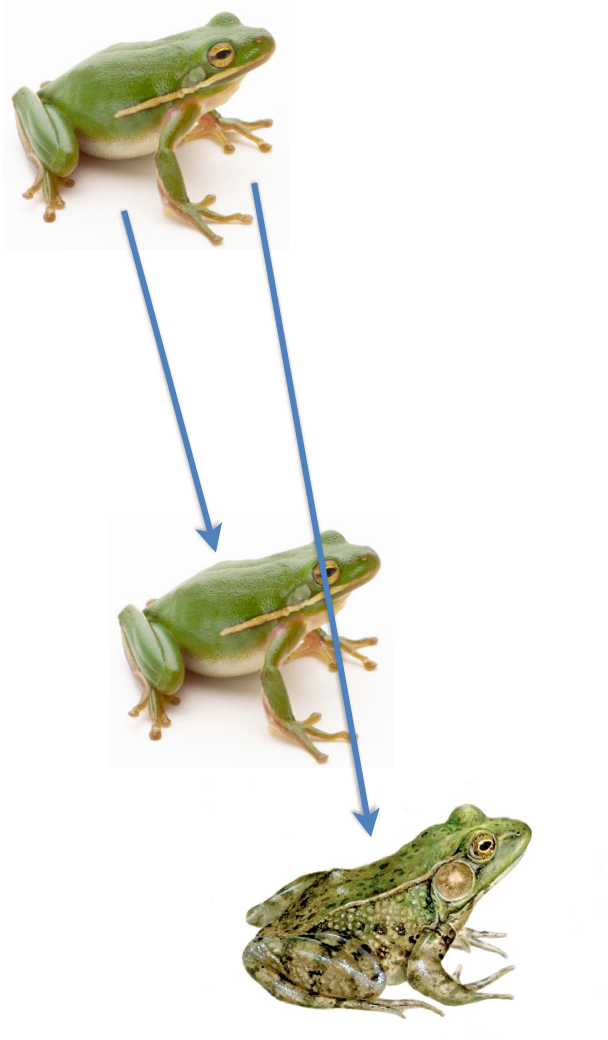
COPY- inside instruments

- In instruments: (see e.g. ILL_H25.instr)

- COMPONENT H25_1 = Guide_gravity(
- w1=0.03, h1=0.2, w2=0.03, h2=0.2, l=L_H25_1,
- R0=gR0, Qc=gQc, alpha=gAlpha, m=m, W=gW)
- AT (0,0,Al_Thickness+gGap) RELATIVE PREVIOUS
- ROTATED (0,Rh_H25_1,0) RELATIVE PREVIOUS

- COMPONENT **MYcopy** = **COPY**(H25_1)
- AT (0,0,L_H25_1+gGap) RELATIVE PREVIOUS
- ROTATED (0,Rh_H25_1,0) RELATIVE PREVIOUS

- COMPONENT **COPY**(H25_1) = **COPY**(H25_1)(W=2*gW)
- AT (0,0,L_H25_1+gGap) RELATIVE PREVIOUS
- ROTATED (0,Rh_H25_1,0) RELATIVE PREVIOUS



GROUP - components working in parallel

AT (0,0,-LMM) RELATIVE Cradle ROTATED (0,A1/2,0) RELATIVE Cradle

***GROUP** IN6Monoks*

AT (0,0,0) RELATIVE Cradle ROTATED (0,A2/2,0) RELATIVE Cradle

***GROUP** IN6Monoks*

- One comp after the particle is “tried” in sequential order until the it becomes SCATTERED.

WHEN

- Syntax:

COMPONENT Mine = Yours(blah, blah)

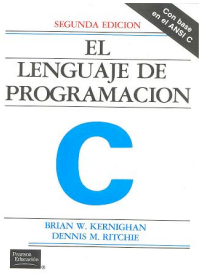
WHEN (c-expression) AT (....)

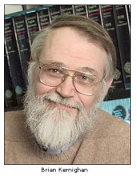
- Is very powerful when combined with EXTEND and user variables, or as a method to let input parameters select if certain components are active.

- Example: Use EXTEND to flag if neutron was scattered on one monochromator blade or another. Then later use WHEN to only show contribution from blade N at sample position?

COMPONENT Mon = PSD_monitor(...)

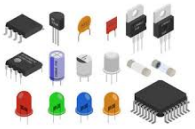
WHEN (myvar==1) AT (0,0,0) RELATIVE Sample




Brian Kernighan


Dennis Ritchie

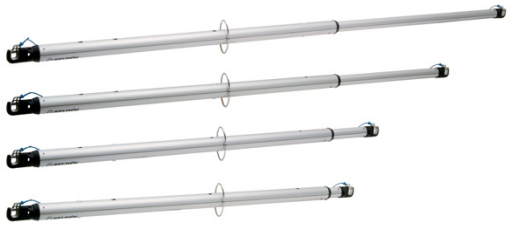
EXTEND



McXtrace

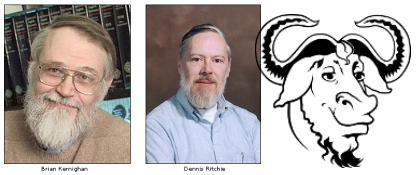
SOLEIL SYNCHROTRON

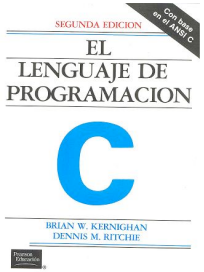
- Enrich component behaviour using EXTEND:



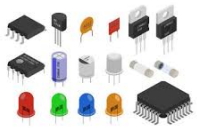
```
COMPONENT Mono1 = Monochromator_curved(...)
AT (0,0, -LMM) RELATIVE Cradle ROTATED (0,A1/2,0) RELATIVE Cradle
GROUP IN6Monoks
EXTEND
%{
  if (SCATTERED) { myvar = 1; }
%}

...
COMPONENT Mono2 = Monochromator_curved(...)
AT (0,0, 0) RELATIVE Cradle ROTATED (0,A2/2,0) RELATIVE Cradle
GROUP IN6Monoks
EXTEND
%{
  if (SCATTERED) { myvar = 2 ;}
%}
```

K & R. / GNU





Combined example: Decompose multiple scattering from Single_crystal



```
DECLARE %{
        double multiple_scatt;
%}
…
COMPONENT Crystal = Single_crystal(… order=0 …)
AT (0,0,0) RELATIVE somewhere
EXTEND %{
        multiple_scatt=SCATTERED;
%}
…
COMPONENT PSD_single=PSD_monitor(…)
WHEN (multiple_scatt==1) AT (0,0,0) RELATIVE somewhere_else

COMPONENT PSD_multiple=PSD_monitor(…)
WHEN (multiple_scatt > 1) AT (0,0,0) RELATIVE somewhere_else
```

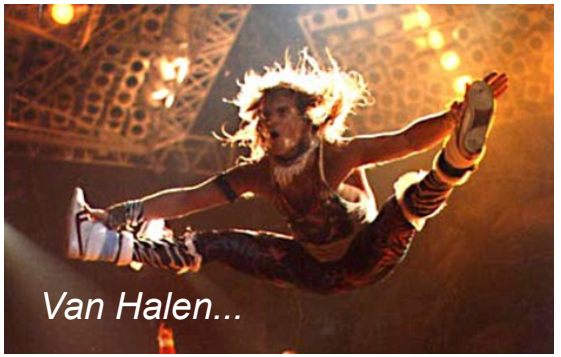
JUMP

- A goto. Be careful. Can be used in two situations:
- **JUMP** to myself
- **JUMP** to an Arm

- No coordinate transformations are applied... (Meaning that if the Arms you JUMP between do not coincide you will “move” / “reorient” the neutrons...)

- Syntaxes:
- COMPONENT a=b(...)
- **WHEN** (expr) AT (...) **JUMP** somewhere

- COMPONENT a=b(...)
- **WHEN** (expr) AT (...) **JUMP** myself



JUMP

- A goto. Be careful. Can be used in two situations:
- **JUMP** to myself
- **JUMP** to an Arr

BEWARE - This IS a GOTO!

- No coordinate transformations are applied... (Meaning that if the Arms you JUMP between do not coincide you will “move” / “reorient” the neutrons...)

- Syntaxes:
- COMPONENT a=b(...)
- **WHEN** (expr) AT (...) **JUMP** somewhere

- COMPONENT a=b(...)
- **WHEN** (expr) AT (...) **JUMP** myself

JUMP

- A goto. Be careful. Can be used in two situations:
- **JUMP** to myself
- **JUMP** to an Arr

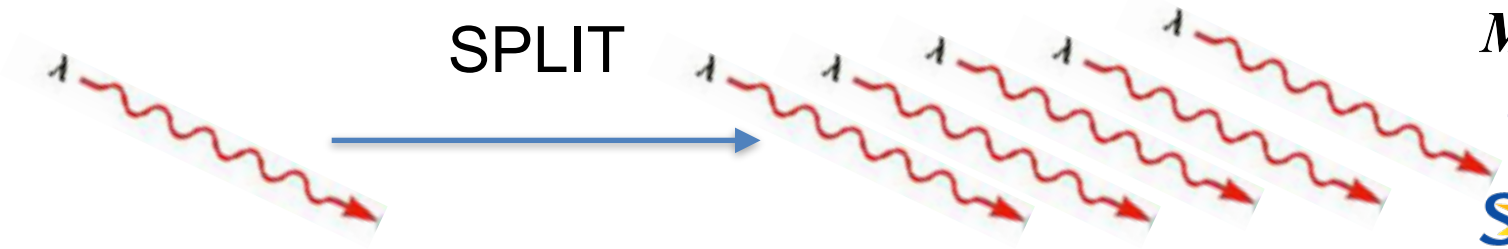
B a GOTO!

- No coordinate trans at if the Arms you JUMP between do not coincide you)

- Syntaxes:
- COMPONENT a=b
- **WHEN** (expr) AT (..

- COMPONENT a=b
- **WHEN** (expr) AT (..

SPLIT

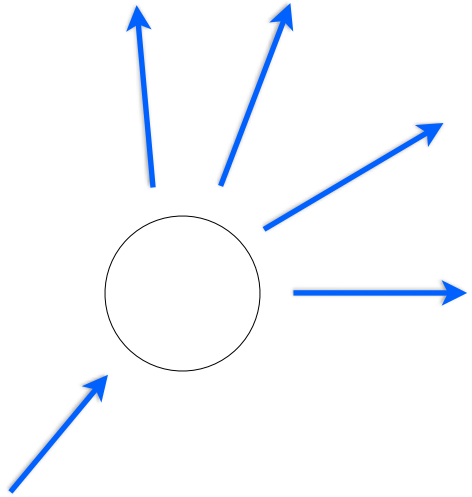


- Increase statistics beyond this point in the instrumentfile

- SPLIT n MyArm = Arm()
- AT somewhere

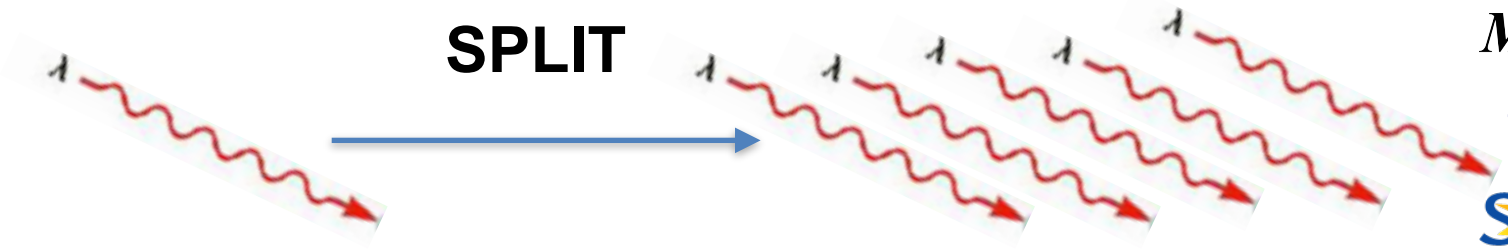
- will “formulate an if-statement”:

- for j=1:n
  - comp1
  - comp2
  - comp3
  - ...
- end (of instrument)



- **ONLY** meaningful in case of Monte Carlo choices after SPLIT point...

SPLIT

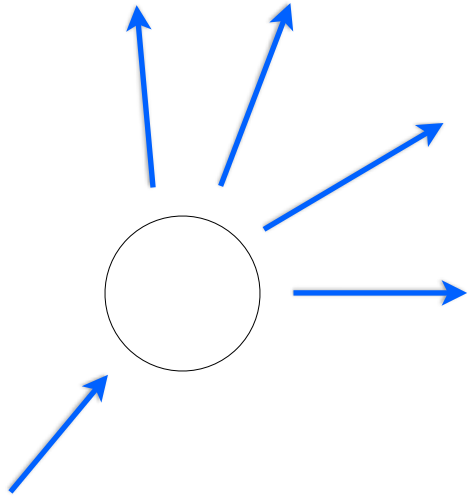


- Increase statistics beyond this point in the instrumentfile

- SPLIT n MyArm = Arm()
- AT somewhere

- will “formulate an if-statement”:

- for j=1:n
  - comp1
  - comp2
  - comp3
  - ...
- end (of instrument)



- **ONLY** meaningful in case of Monte Carlo choices after SPLIT point...

Works very well together with e.g. monochromators, Single_crystal, PowderN