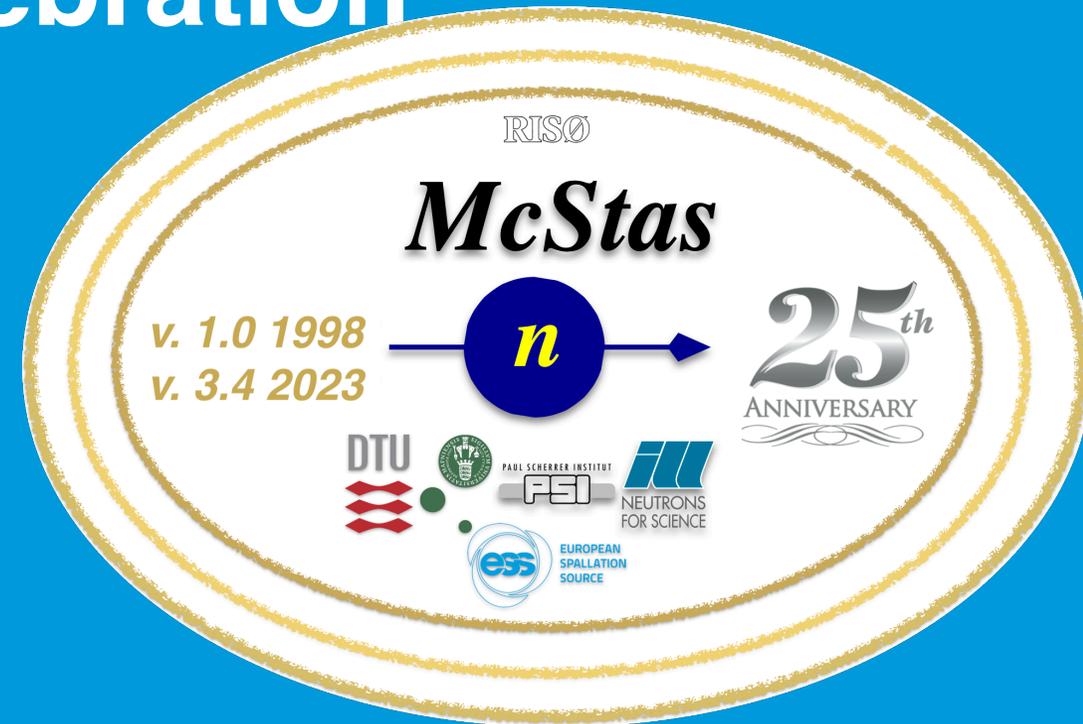


McStas 25 year celebration

2023/11/16

PETER WILLENDRUP, ESS DMSC & DTU PHYSICS

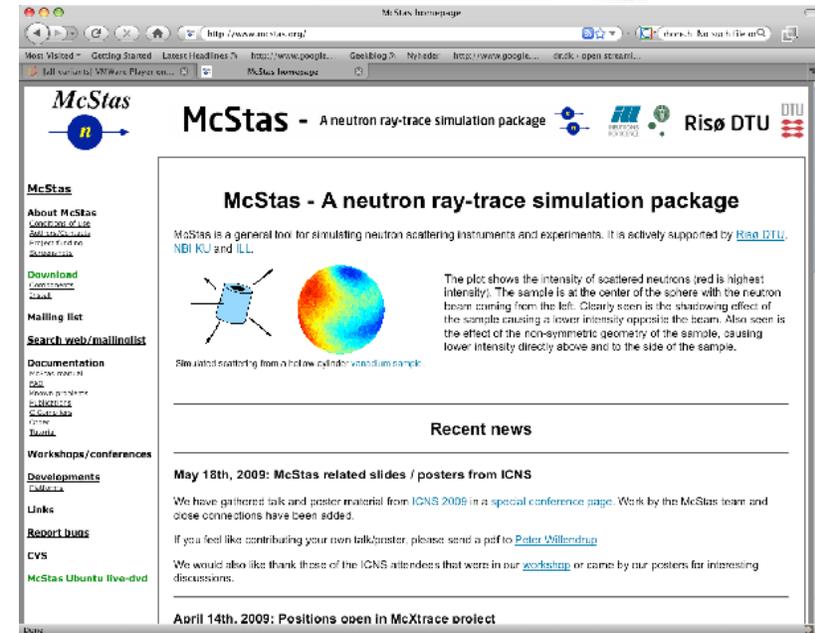


McStas Introduction

- Flexible, general simulation utility for neutron scattering experiments.
- Original design for **Monte carlo Simulation of triple axis spectrometers**
- Developed at DTU Physics, ILL, PSI, Uni CPH, ESS DMSC
- V. 1.0 by **K Nielsen & K Lefmann** (1998) RISØ
- Currently ~6 people on joint McStas-McXtrace team but only **2 full time**, based at ESS DMSC / DTU



GNU GPL v3 license
Open Source



Project website at

<http://www.mcstas.org>

mcstas-users@mcstas.org mailinglist

Main Page - McXtraceWiki

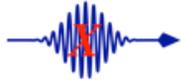
http://www.mcxtrace.org/index.php?title=Main_Page

Google

Most Visited Getting Started Latest Headlines http://www.google... Geekblog Nyheder http://www.google... dr.dk open streami...

Log in / create account

McXtrace



article discussion edit history

Main Page

McXtrace

McXtrace - Monte Carlo Xray ray-tracing is a joint venture by



Funding from NABIIT, DSF and the above parties.

Our code will be based on technology from .

For information on our progress, please subscribe to our [user mailinglist](#).

<mailto:webmaster@mcxtrace.org>

navigation

- Main Page
- Partners
- Project People
- Project Status
- Vacancies
- Project Goal
- Mailing List
- Links
- SMEXOS talks
- SRI09 abstracts

search

Go Search

toolbox

- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link

This page was last modified 13:15, 25 February 2009.

This page has been accessed 2,049 times.

[Privacy policy](#)

[About McXtraceWiki](#)

[Disclaimers](#)



- Synergy, knowledge transfer, shared infrastructure, repo etc.

Funding sources

During its lifetime, McStas efforts have been supported through several European Union RTD and JRA programmes, plus several instrument development projects for facilities.

Project	Program type	Funding period
XENNI	RTD (EU FP4)	1996 - 2000
Cool Neutrons	RTD (EU FP4)	1998 - 2001
SCANS	RTD (EU FP5)	2000 - 2004
MCNSI	JRA in NMI3 (EU FP6)	2004 - 2006
MCNSI7	JRA in NMI3/FP7 (EU FP7)	2006 - 2008
NMI3-II/FP7 outreach project	JRA in NMI3/FP7 (EU FP7)	2012 - 2016
ISIS TS2 EU project	Infrastructure project in (EU FP6)	2006-2009
Instrument simulations for the ESS design update	Danish in-kind project toward the ESS	2009-2012
Secondment of P Willendrup from DTU for supporting instrument simulations for the ESS	33% part of the ESS DMSC	2014-2022
SINE2020	Part of WP3 and WP8 (EU H2020)	2016 - 2019
PaNOSC	Part of Software and E-learning (EU H2020)	2019 - 2022
HighNESS	Contributions to WPs 6,8,9 (EU H2020)	2020 - 2023
Secondment of P Willendrup from DTU for supporting instrument simulations for the ESS	100% part of the ESS DMSC	2023-2027



McStas: simulation toolkit for neutron scattering instruments, virtual experiments

2020: McStas 3.0 with support for MPI and multiple GPUs

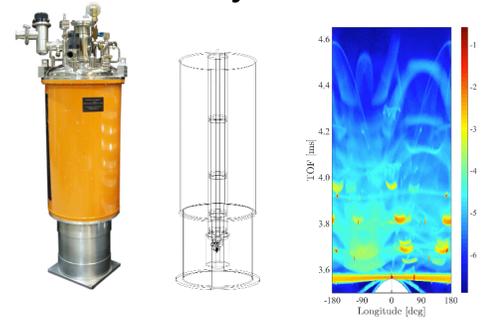
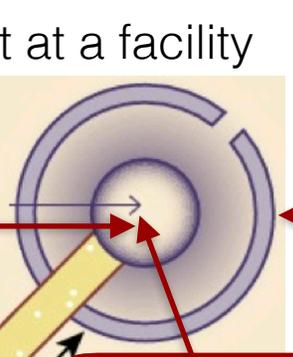
McStas + 



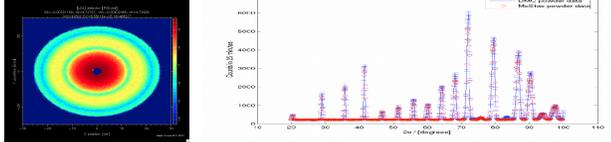
McStas 3.x acc. on NVIDIA GPUs

- 2 orders of magnitude speedup.
(1x Tesla V100 vs 1 modern Intel Xeon core)

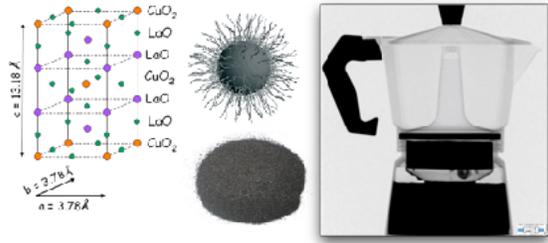
Sample-environments - Union subsystem

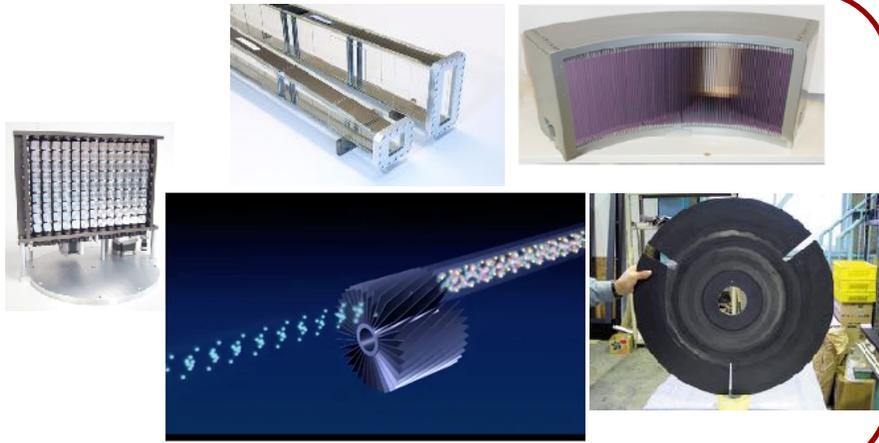
Detectors



Scientific model-samples



Neutron optics



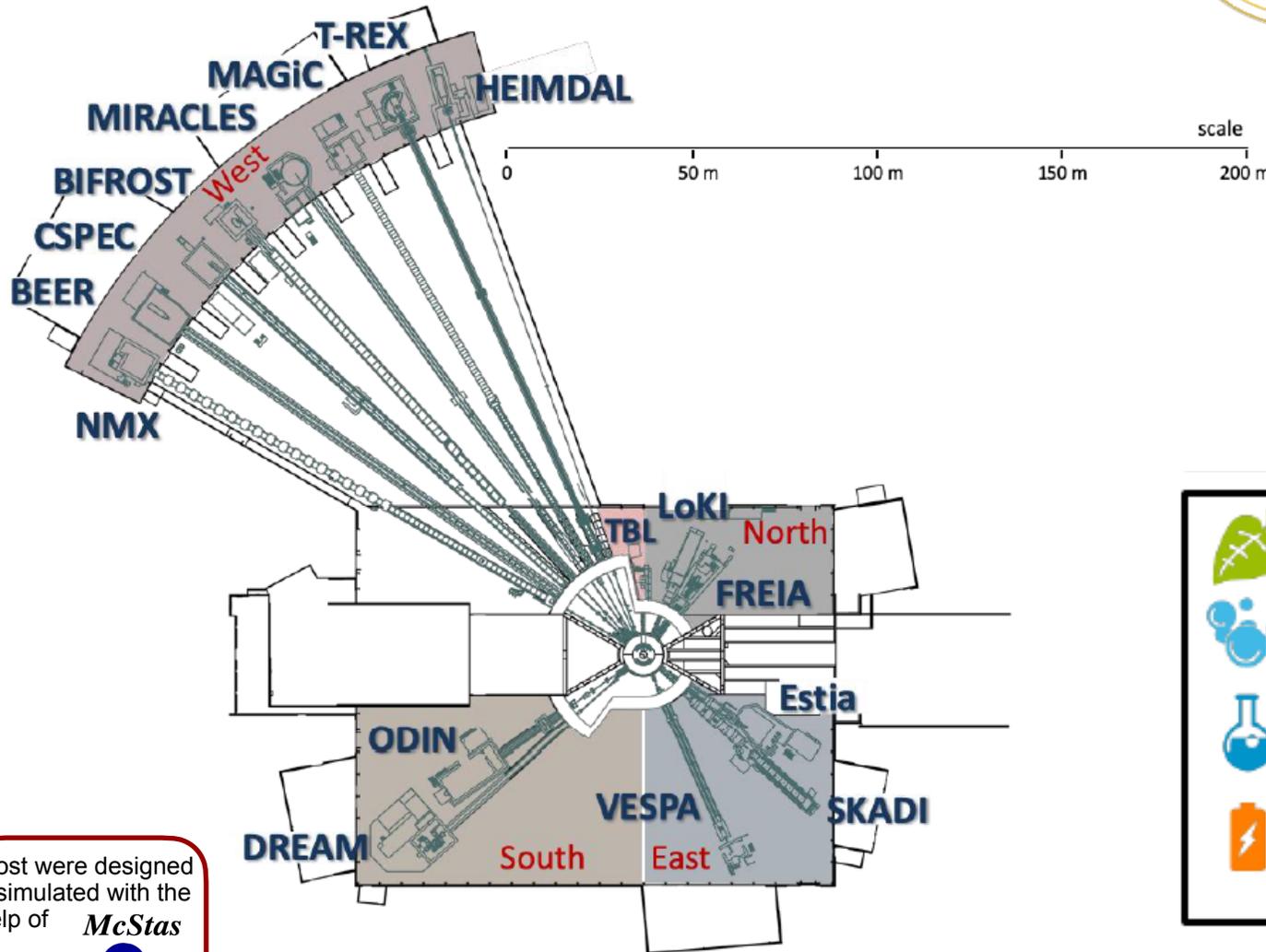
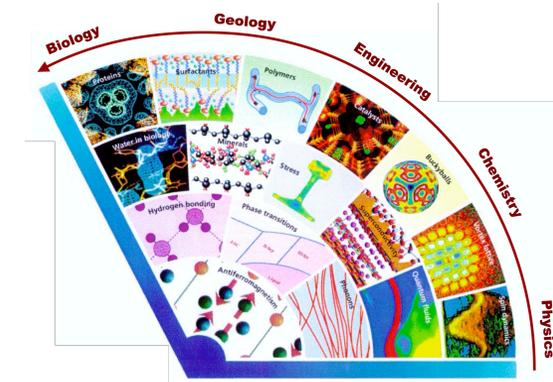
Starts with a source of neutrons, be it a reactor- or spallation source

Neutron moderators is where  starts

Used in many places



Instrument suite at ESS



15 instruments

5 Large-scale structure instruments

5 Diffractometers

5 Spectrometer

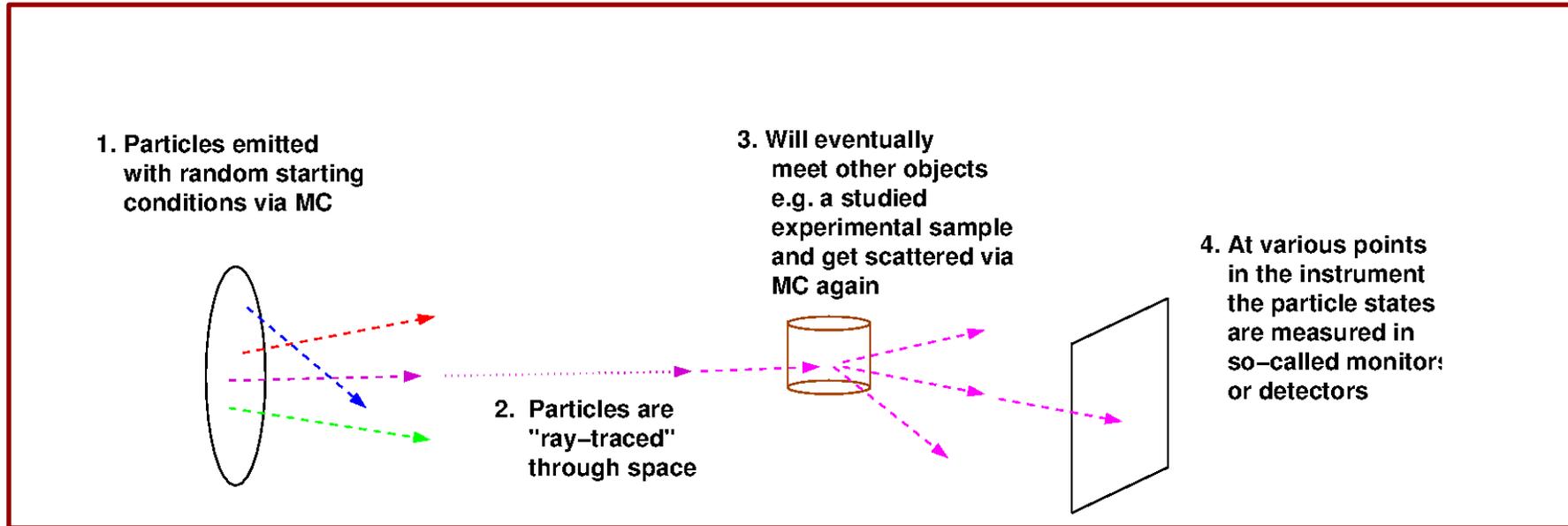
	life sciences		magnetism & superconductivity
	soft condensed matter		engineering & geo-sciences
	chemistry of materials		archeology & heritage conservation
	energy research		particle physics

Most were designed / simulated with the help of **McStas**

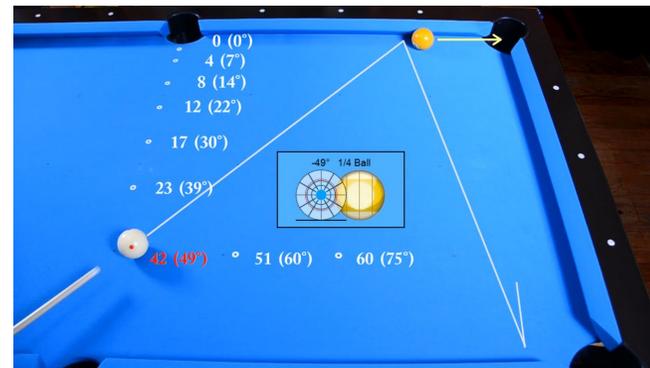
:-)



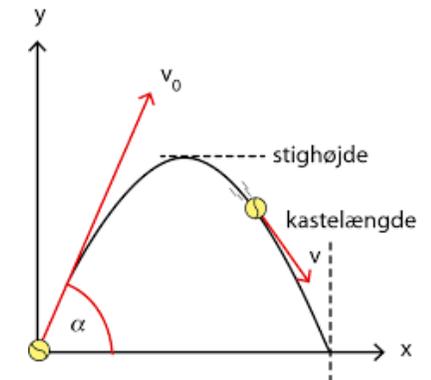
McStas and McXtrace are Monte Carlo ray-tracers



- For the neutrons, gravity kicks in... A cold neutron falls ~10cm over 150m!
- Classical Newtonian mechanics, i.e.
- (independent, particles though...)



d



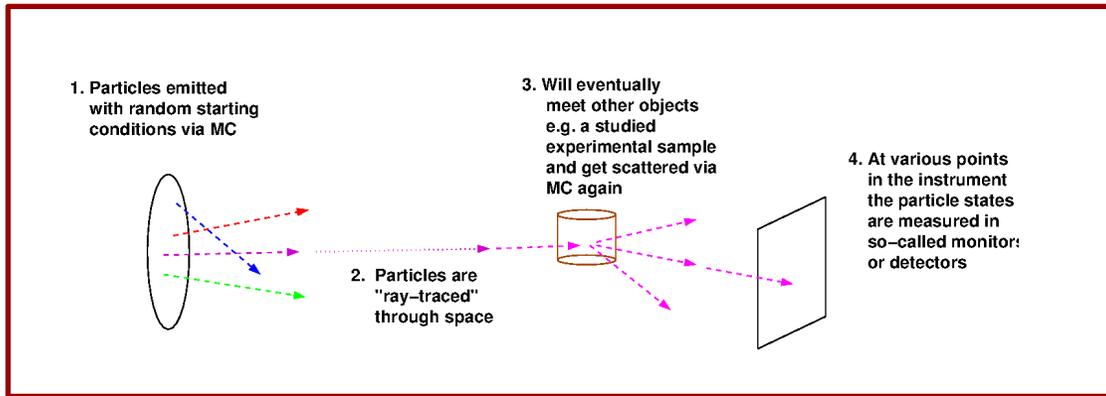
McXtrace



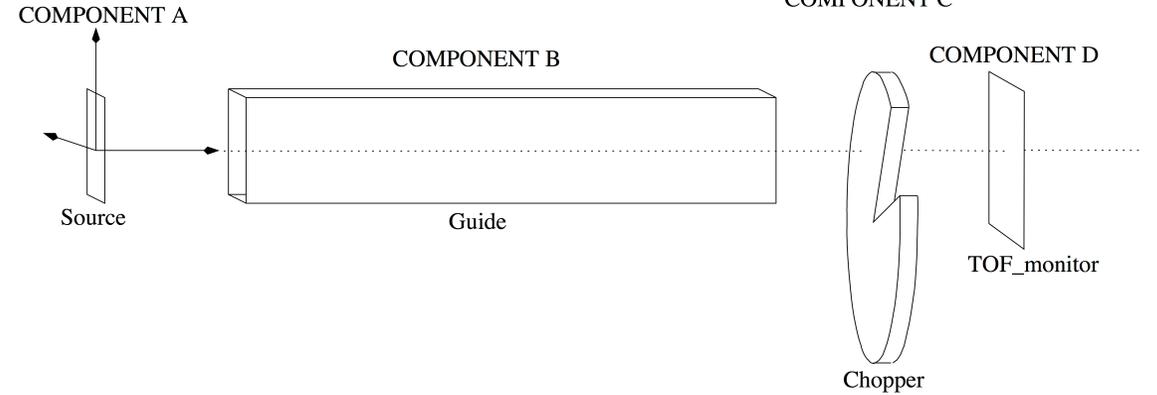
McStas



McStas and McXtrace Monte Carlo ray-tracers



INSTRUMENT



The "tool layer" consists of programs manipulated by the McStas user:

mcgui, graphical user interface

mcplot, visualize histogram outp.

mcdisplay, visualize instrument

mcgui is used to assemble an instrument file, which is taken over by the McStas system

DEFINE INSTRUMENT Example(Param1=1, string Param2="two", ...)

COMPONENT A = Source(Parameters...)
AT (0, 0, 0) ABSOLUTE

COMPONENT B = Guide(Parameters...)
AT (0, 0, 1) RELATIVE A

COMPONENT C = DiskChopper(Parameters...)
AT (0, 0, 1) RELATIVE B

COMPONENT D = TOF_monitor(Parameters, filename="Tof.dat")
AT (0, 0, Param1) RELATIVE PREVIOUS

"Instrument file"

DSL

Source.comp – c-code

Guide.comp – c-code

DiskChopper.comp – c-code

TOF_monitor.comp – c-code

Component library

Random numbers

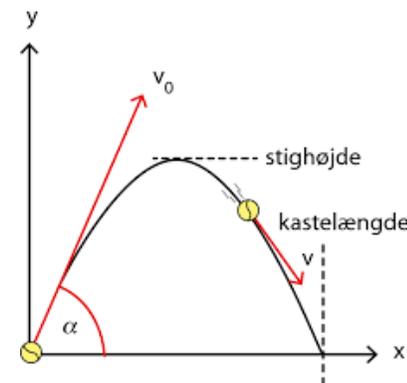
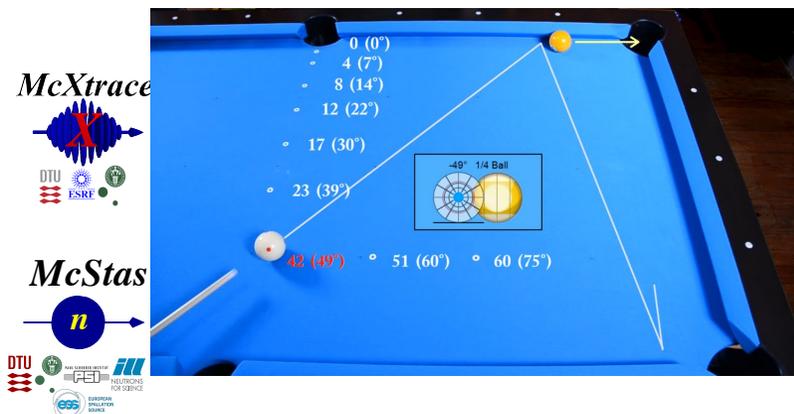
I/O

Physical consts.

"Kernel and runtime c-code"

The McStas system generates an "ISO C file" and an executable from instrument file and c-codes

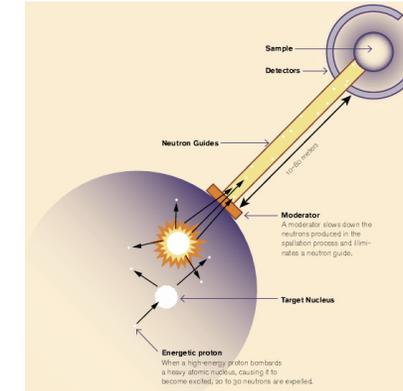
The simulation executable produces data output which can be visualized using the mcplot and mcdisplay tools



McStas tech overview

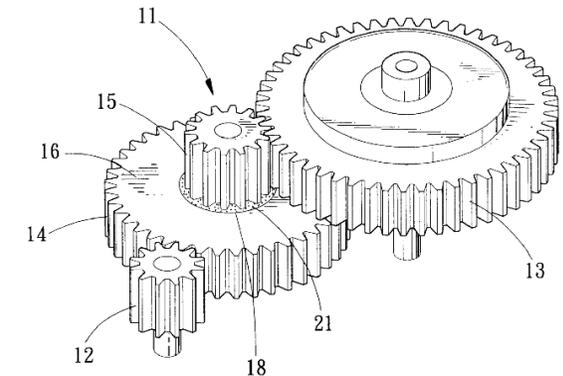


- Portable code (Unix/Linux/Mac/Windows)
 - On the CPU-side, ran on everything from iPhone to 1000+ node cluster, intel, Alpha, PA-RISC etc.
- 'Component' files (>300) inserted from library
 - Sources
 - Optics
 - Samples
 - Monitors
 - If needed, write your own comps - **many are USER developments ~200-line "physicist" codes**



- DSL + ISO-C code-gen. (compiler technology / LeX+Yacc)
 - Simple Instrument language $\xrightarrow{\text{Code generation}}$ ISO C

- Component codes realizing beamline parts (including user contribs)



- Library of common functions

- I/O
- Random numbers
- Physical constants
- Propagation
- Precession in fields
- ...

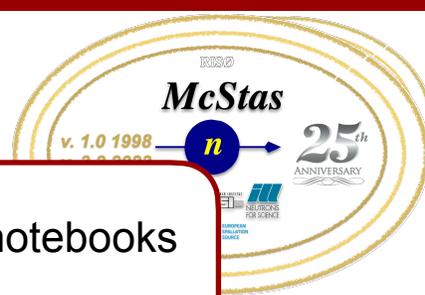
User experience:

- Write instrument
- Launch simulation (generates binary and runs simulation)
- Look at output data

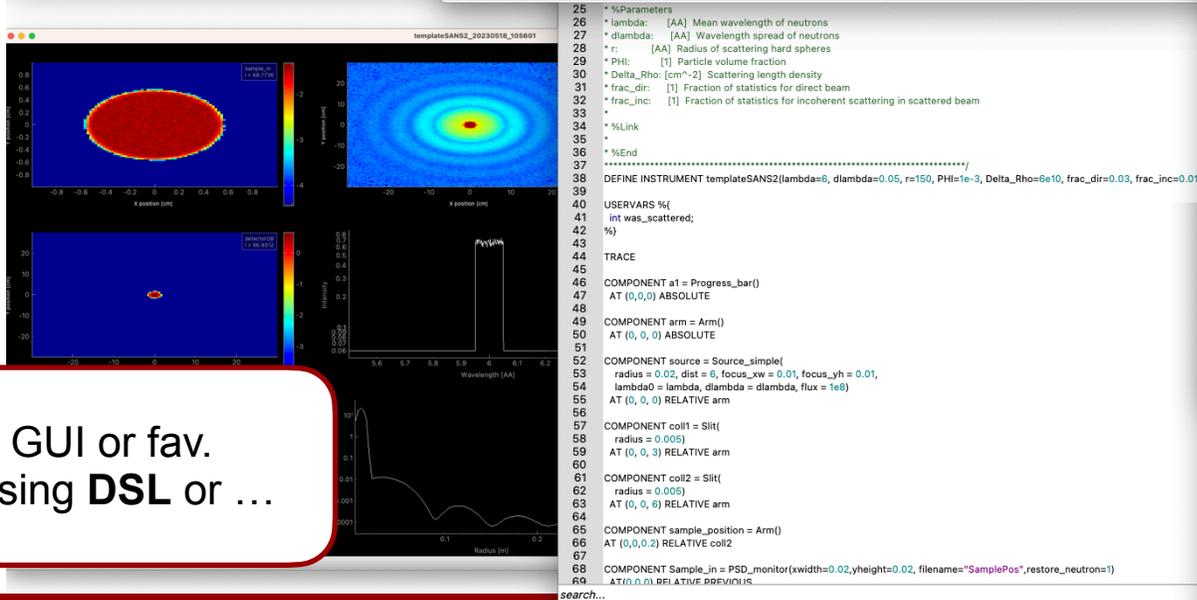
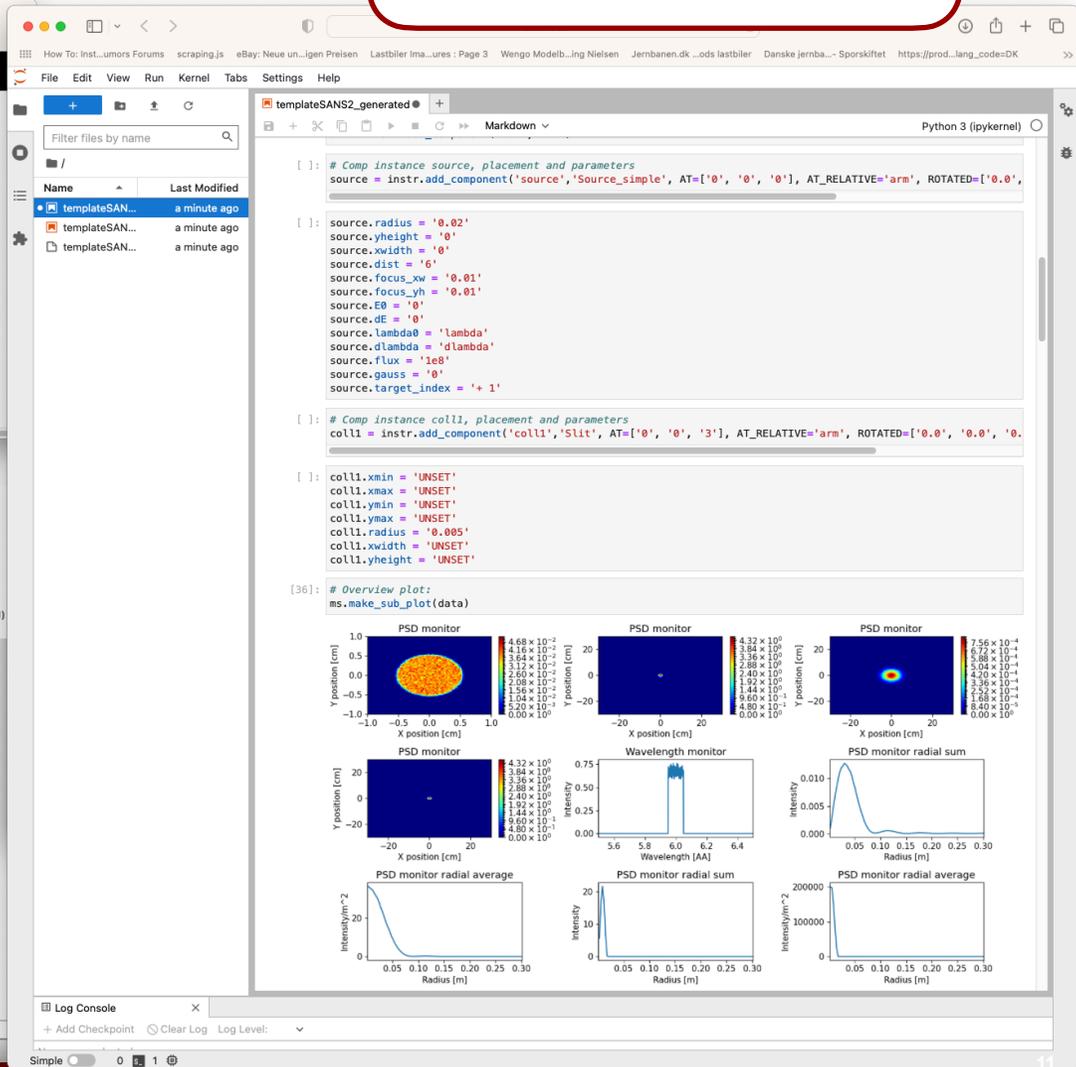
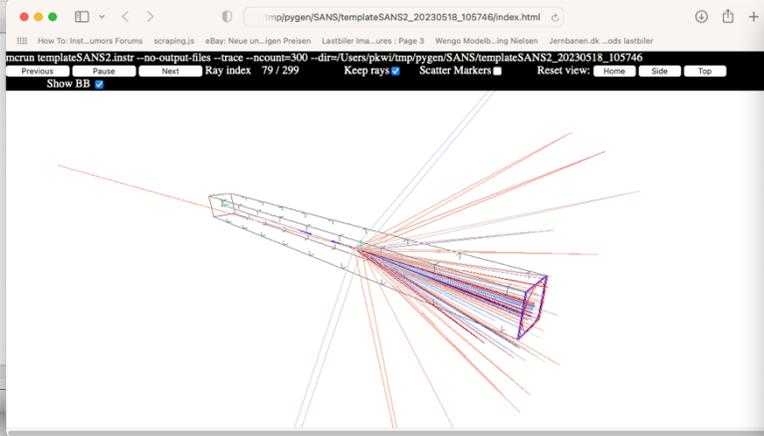
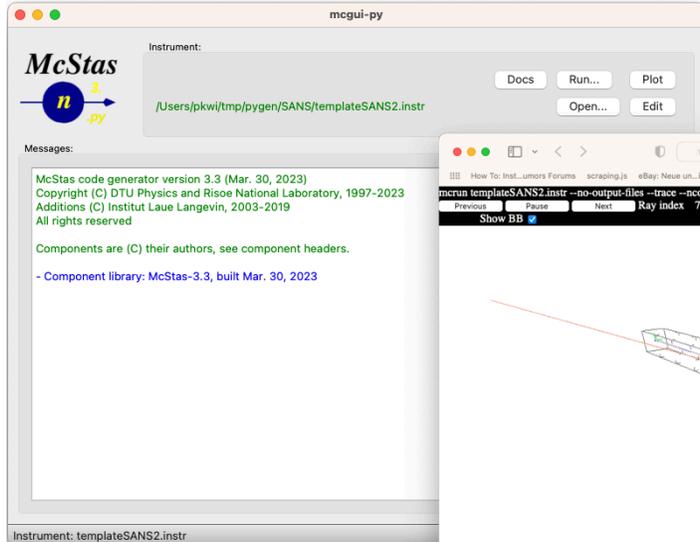




McStas: simulation toolkit for neutron scattering instruments, V.E.



... in .py / Jupyter notebooks using McStasscript



Work in GUI or fav. editor using DSL or ...

McStas is by design a “linear chain” of components

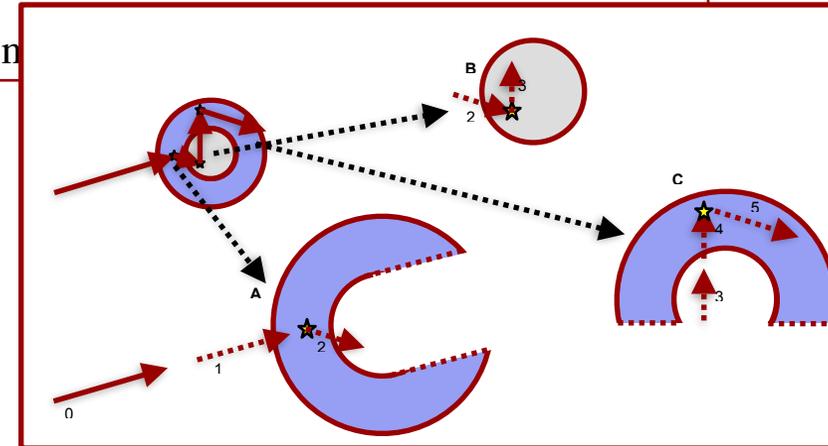
- But:
 - We have syntaxes/logic to e.g. GROUP components. (Think: XOR and similar logic)

```
{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] }
```

McStas is by design a “linear chain” of components

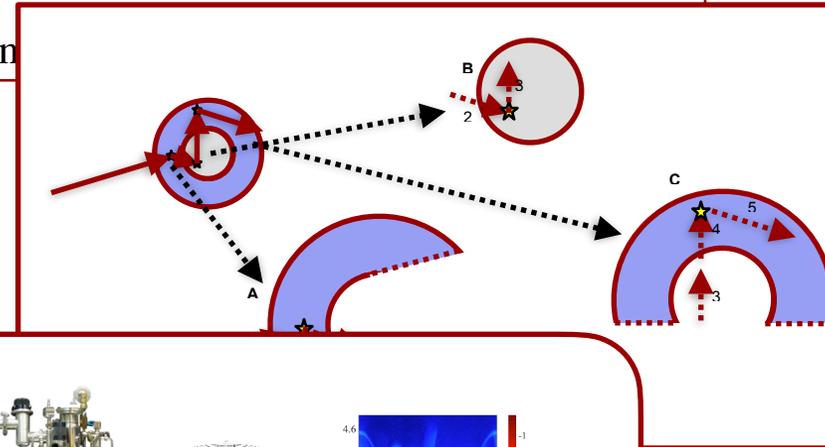
- But: `{SPLIT} COMPONENT name = comp(parameters) {WHEN condition}`
- We can: `AT (...) [RELATIVE [reference|PREVIOUS] | ABSOLUTE]`
- `{ROTATED {RELATIVE [reference|PREVIOUS] | ABSOLUTE} }`
- `{GROUP group_name}`
- `{EXTEND C_code}`
- `{JUMP [reference|PREVIOUS|MYSELF|NEXT] [ITERATE n]}`

- Material-assemblies may be arranged in “concentric” onion-shells (Single_crystal, PowderN, ...)

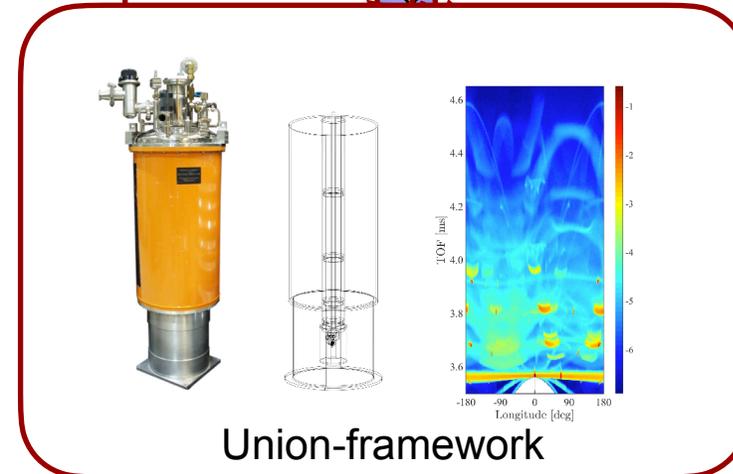


McStas is by design a “linear chain” of components

- But: `{SPLIT} COMPONENT name = comp(parameters) {WHEN condition}`
- We can: `AT (...) [RELATIVE [reference|PREVIOUS] | ABSOLUTE]`
- `{ROTATED {RELATIVE [reference|PREVIOUS] | ABSOLUTE} }`
- `{GROUP group_name}`
- `{EXTEND C_code}`
- `{JUMP [reference|PREVIOUS|MYSELF|NEXT] [ITERATE n]}`



- Material-assemblies may be arranged in “concentric” onion-shells (Single_crystal, PowderN, ...)
- The Union subsystem (Mads Bertelsen) has been added, defining region(s) of the instrument where geometry and materials are decoupled and we completely deviate from the linear approximation
- NCrystal may be used to describe materials, also within Union. `cfg=“materials_galore.ncmat”`



McStas is by design a “linear chain” of components

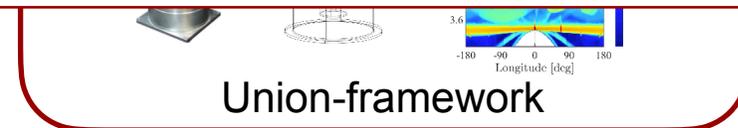
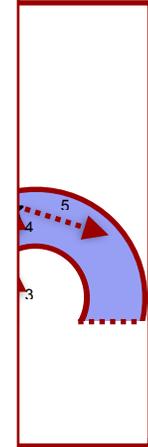
- But:
 - We can use “CPU” to label non-GPU execution of a component in mixed CPU/GPU setups
 - “SHELL” for cogen/compile-time system-hacks (create an instr snippet using e.g. bash)
- Material:


```
SEARCH "/the/path/to/add/" COMPONENT ...
SEARCH SHELL "the_executable --and --some --options" COMPONENT ...

COMPONENT Origin = Progress_bar()
    AT (0,0,0) ABSOLUTE
```
- The instrument can be defined from a Python function:

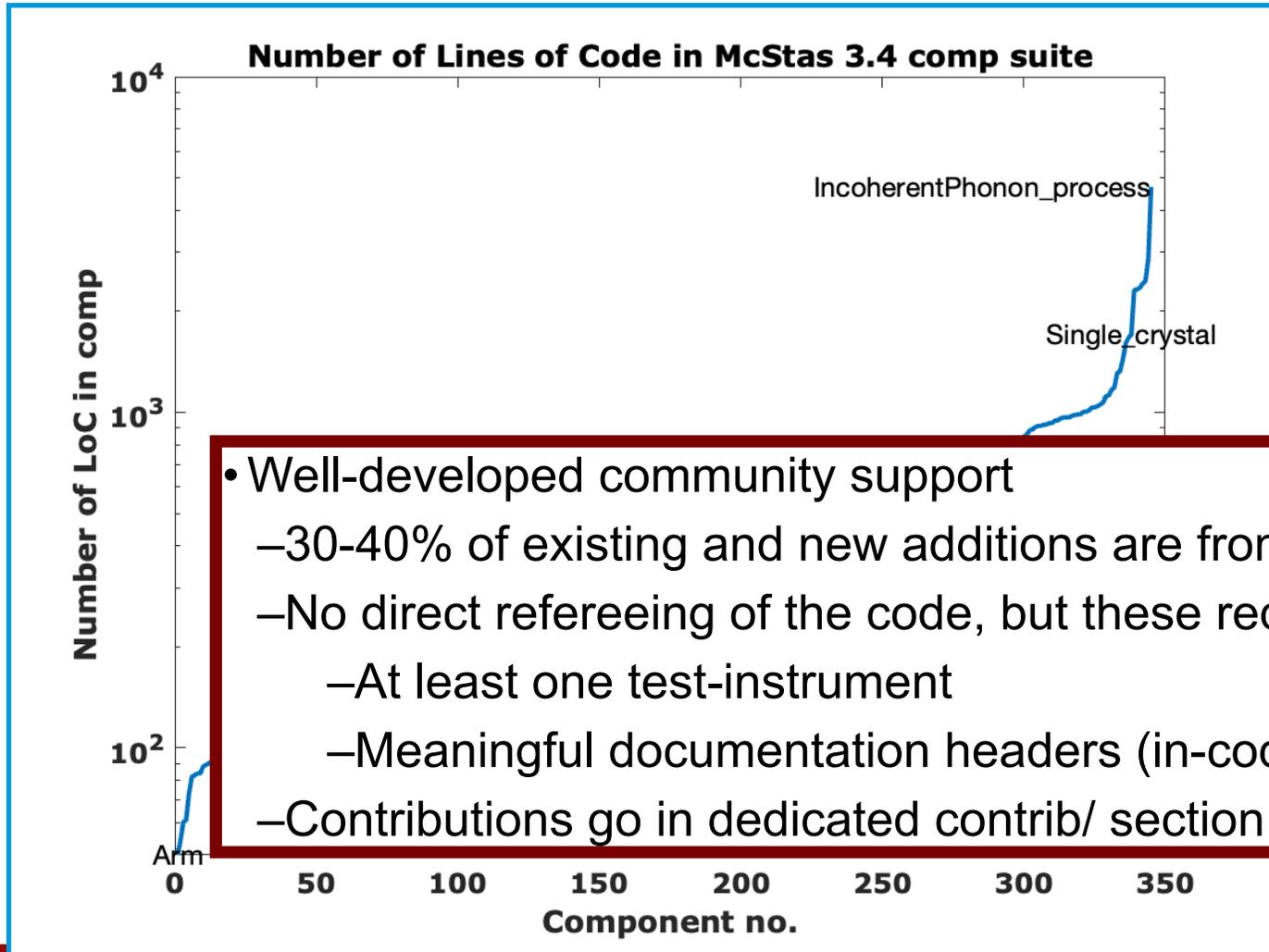

```
METADATA JSON eniuis_data %{
    ...{JSON string}...
%}

METADATA Python extra_function %{
    def the_function(a, b, c):
        return a + b / c
%}
```
- NCrystal may be used to describe materials, also within Union. `cfg="materials_galore.ncmat"`



Writing new comps or understanding existing is not so complex...

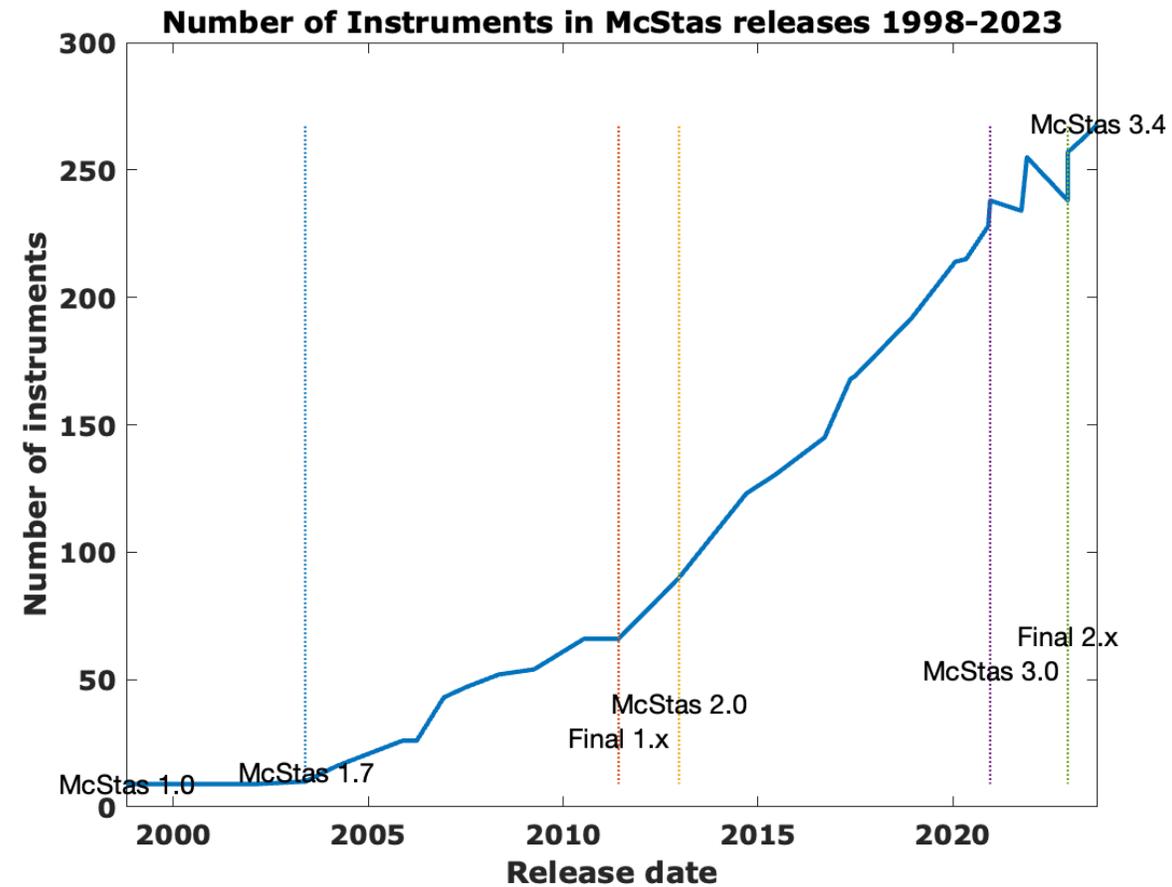
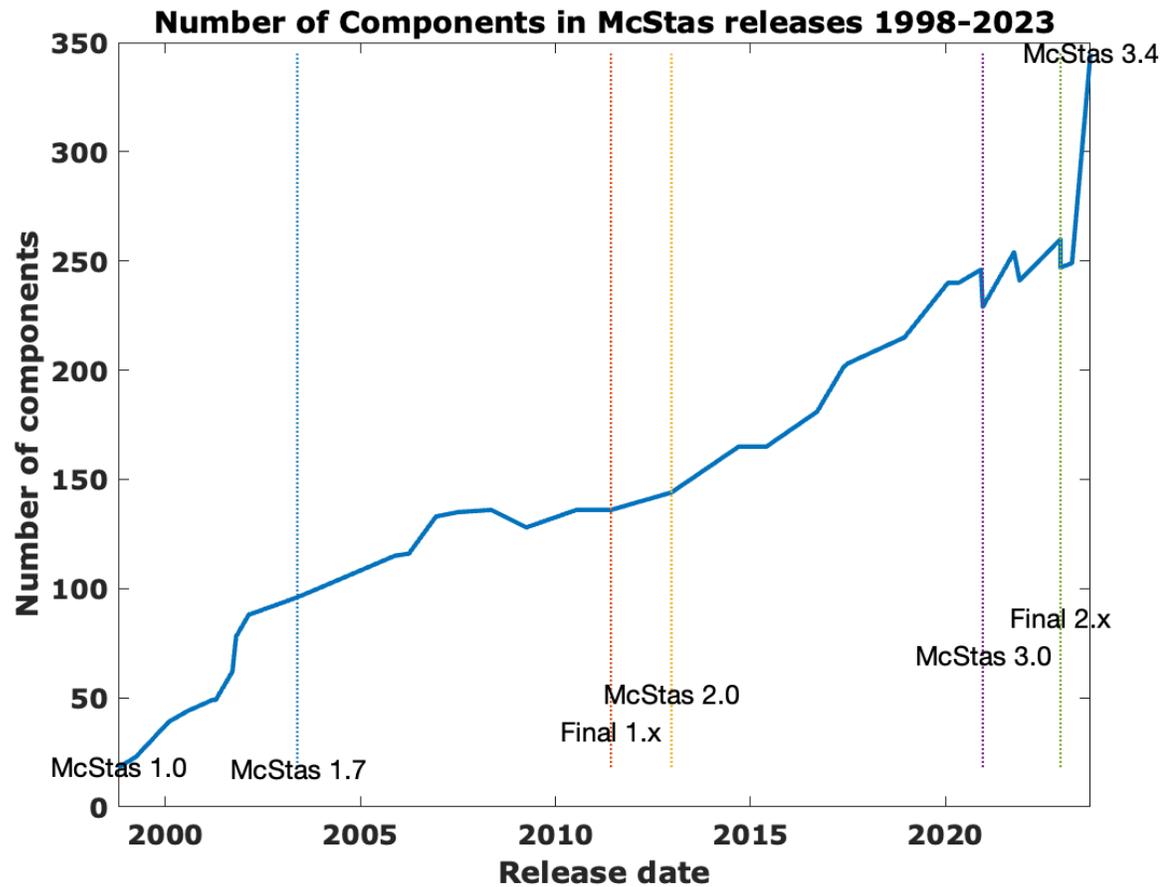
- Many of them are quite simple and short... Statistics:



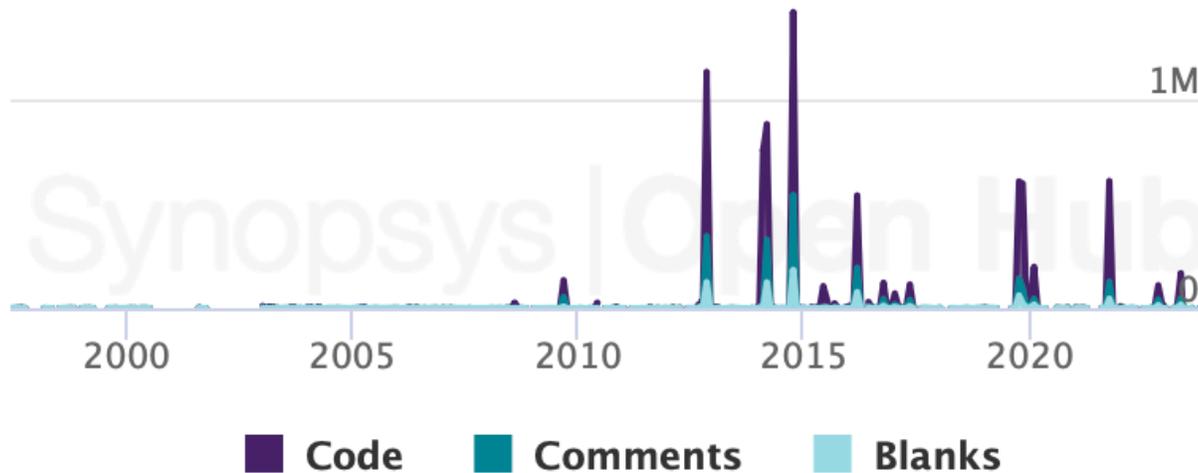
- Well-developed community support
 - 30-40% of existing and new additions are from users
 - No direct refereeing of the code, but these requirements:
 - At least one test-instrument
 - Meaningful documentation headers (in-code docs)
 - Contributions go in dedicated contrib/ section of library

Component lib development

More than 350 comps, more than 250 instr examples



Lines of Code



In a Nutshell, McCode...

- ... has had 24,640 commits made by 60 contributors representing 2,177,413 lines of code
- ... is mostly written in Postscript with a low number of source code comments
- ... has a well established, mature codebase maintained by a large development team with increasing Y-O-Y commits
- ... took an estimated 636 years of effort (COCOMO model) starting with its first commit in June, 1997 ending with its most recent commit about 1 month ago

Languages

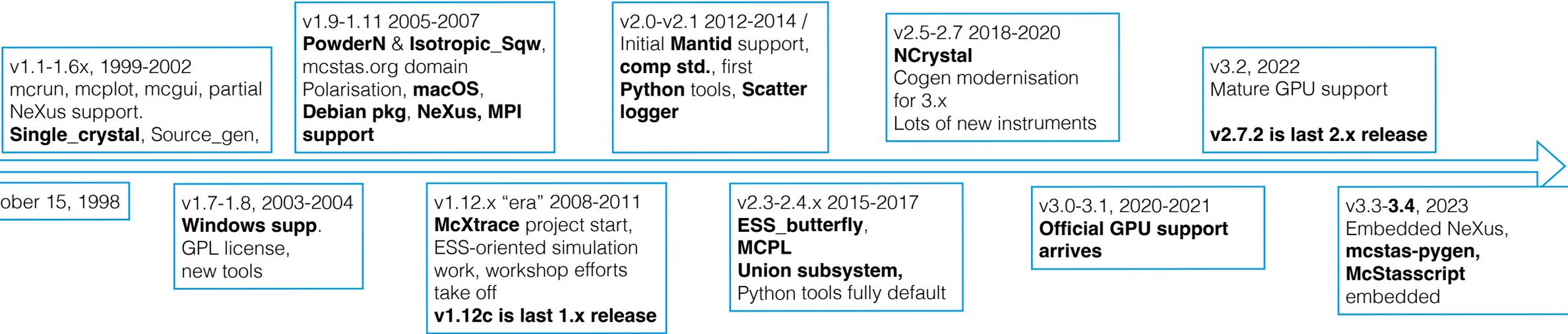


Nope, that's our DSL and grammar. :-) Which is close to "English".



McStas timeline at a glance

When did what functionality arrive



Steady in-flow of “smaller” developments, bugfixes, user contributions...

McStas collaboration



Thanks and acknowledgments of people...

Kristian Nielsen
 Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind “internals” and LeX-Yacc gram.
 Releases 1.0-1.4



Kim Lefmann
 Author of many components, orig. “power user”, supervised many KU stud. and secured funding.
 Part of McStas all along the way.

Henrik Rønnow
 Drafted v 1.0 grammar w. Kim & Kristian



Kurt Clausen:
 Came up with original idea of “framework”, secured first EU funding.



v. 1.0 1998
 v. 3.4 2023



McStas



25th
 ANNIVERSARY

Emmanuel Farhi

First non-DK “McStas system developer”,



PW mentor for the deeper tech.

Grammar ext. advanced comps. + lots more
 Now works mainly on McXtrace.

McStas collaboration

Thanks and acknowledgements



Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind “internals” and LeX-Yacc gram.
Releases 1.0-1.4

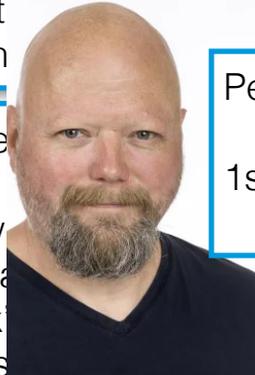


Kim Leifer
Author of compiler
“power” supervisor
KU student security



PW:
“Mr. McStas”
2002- ?
Custodian, community caretaker, sustained workforce.
Support for newcomers and power users alike.

Part along

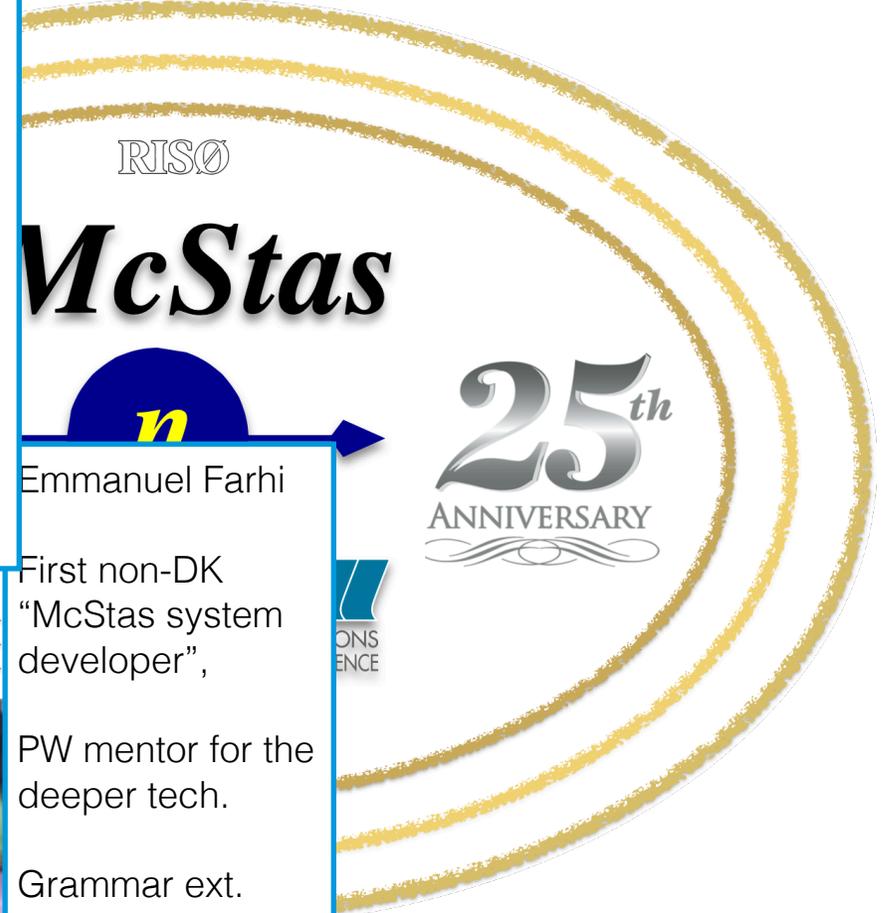


Peter Christiansen
1st pol. solution

Kurt Clausen
Came up with original idea
“framework” secured first funding.



Henrik Rønnow
Drafted v 1.0 grammar w. Kim & Kristian



Emmanuel Farhi
First non-DK “McStas system developer”,
PW mentor for the deeper tech.



Grammar ext. advanced comps. + lots more
Now works mainly on McXtrace.

McStas collaboration

Thanks and acknowledgements



Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind “internals” and LeX-Yacc gram.
Releases 1.0-1.4



Kim Le...
Author compo...
“power superv...
KU stu...
secure...
Part alon...



PW:
“Mr. McStas” 2002- ?
Custodian, community caretaker, sustained workforce.
Support for newcomers and power users alike.

Mads Bertelsen
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas:
guide_bot and Union systems.



Kurt Clause...
Came up w...
original idea...
“framework...
secured first...
func...



Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more
Left for CPH Atomics in 2022.



Peter Christiansen
1st pol. solution

Emmanuel Farhi
First non-DK “McStas system developer”,
PW mentor for the deeper tech.

Grammar ext. advanced comps. + lots more
Now works mainly on McXtrace.



McStas collaboration

Thanks and acknowledgements



Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind "internals" and LeX-Yacc gram.
Releases 1.0-1.4



Kim Leung
Author of compiler, "power" supervisor, KU student security



PW:
"Mr. McStas" 2002- ?
Custodian, community caretaker, sustained workforce.
Support for newcomers and power users alike.



Esben Klinkby
Mcnp-hooks, scatter-logger

Mads Bertelsen
"Next generation", Phd with KL, contributed adv. systems "around" and "in" McStas:
guide_bot and Union systems.



Henrik Rønnow

Kurt Clausen
Came up with original idea "framework" secured first funding



Peter Christiansen
1st pol. solution

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more
Left for CPH Atomics in 2022.



Jakob Garde
2nd round of .py tools, GPU deeper tech.

Grammar ext. advanced comps. + lots more
Now works mainly on McXtrace.



Johan Brinch
Transfer to CMake 1st round of .py tools

25th

McStas collaboration

Thanks and acknowledgements

Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind "internals" and LeX-Yacc gram.
Releases 1.0-1.4



Kim Leung
Author of compiler, "power supervisor" KU student security



PW:
"Mr. McStas" 2002- ?
Custodian, community caretaker, sustained workforce.
Support for newcomers and power users alike.



Esben Klinkby
Mcnp-hooks, scatter-logger



Torben R. Nielsen:
Contributed solution for connection with "Mantid" data reduction, work on SASmodel integration



Gregory Tucker
ESS event-formation hookup, recent syntax / code-generator additions

Mads Bertelsen
"Next generation", Phd with KL, contributed adv. systems "around" and "in" McStas:
guide_bot and Union systems.



Kurt Clausen
Came up with original idea "framework" secured first funding



Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more
Left for CPH Atomics in 2022.

Peter Christiansen
1st pol. solution



Jakob Garde
2nd round of .py tools, GPU deeper tech.



Johan Brinch
Transfer to CMake 1st round of .py tools

Thomas Kittelmann
Main developer of MCPL particle list Format and NCrystal structure/dynamics lib for MC



McStas collaboration

Thanks and acknowledgements

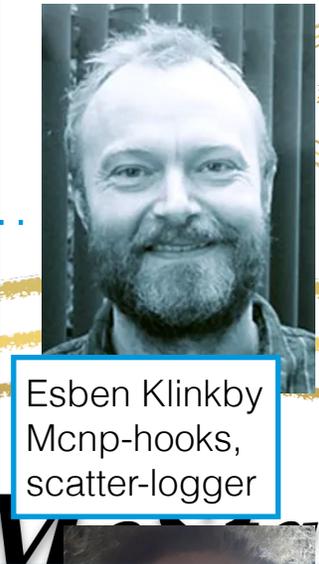
Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind "internals" and LeX-Yacc gram.
Releases 1.0-1.4



Kim Leung
Author of compiler "power" supervisor
KU student security



PW:
"Mr. McStas" 2002- ?
Custodian, community caretaker, sustained workforce.
Support for newcomers



Esben Klinkby
Mcnp-hooks, scatter-logger



Torben R. Nielsen:
Contributed solution for connection with "Mantid" data reduction, work on SASmodel integration



Gregory Tucker
ESS event-formation hookup, recent syntax / code-generator additions

Plus MANY others among the user community at neutron-facilities, students etc.

Mads Bertelsen
"Next generation", Phd with KL, contributed adv. systems "around" and "in" McStas:
guide_bot and Union systems.



behind McXtrace. + lots more
Left for CPH Atomics in 2022.



Grammar ext. advanced comps. + lots more
Now works mainly on McXtrace.



Johan Brinch
Transfer to CMake 1st round of .py tools



Thomas Kittelmann
Main developer of MCPL particle list Format and NCrystal structure/dynamics lib for MC

Demo time?? :-)

