Instrument Modeling: Monte Carlo Methods

Andrew Jackson
European Spallation Source

XIV School of Neutron Scattering "Francesco Paolo Ricci" (SoNS)
2nd Course of the Erice School "Neutron Science And Instrumentation"
"Designing And Building A Neutron Instrument**"**

Erice 1st – 9th April 2016

with Thanks to Peter Willendrup, DTU & ESS

McStas from an Instrument Scientist Perspective

Andrew Jackson
European Spallation Source

XIV School of Neutron Scattering “Francesco Paolo Ricci” (SoNS)
2nd Course of the Erice School “Neutron Science And Instrumentation”
“Designing And Building A Neutron Instrument**”**

Erice 1st – 9th April 2016

with Thanks to Peter Willendrup, DTU & ESS

Why do simulations?

- Cross check analytical results
- Consider configurations/geometries that are hard/ impossible to calculate analytically
- Examine experimentally inaccessible quantities
- Predict instrument performance
- Design experiments

Why do simulations?

- Cross check analytical results
- Consider configurations/geometries that are hard/ impossible to calculate analytically
- Examine experimentally inaccessible quantities
- Predict instrument performance
- Design experiments

Not a substitute
for
thinking!

What are Monte Carlo methods?

Used by Nature since ... (a long time) : diversity of Life

First application using computers:

Metropolis, Ulam and Von Neumann at Los Alamos, 1943

Neutron Scattering and Absorption in *U* and *Pu*, Origin of *MCNP*

Name:

Monte Carlo casino, a random generator (Ulam's father played poker)

What are Monte Carlo Methods

- Use random generators
- Explore a complex and large phase space (many parameters)
- Integrates microscopic random events into measurable quantities **not** a usual regular sampling integration

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1,\, a<u_i<b}^{n} f(u_i) = \frac{1}{b-a} \int_a^b f(u)\,du$$

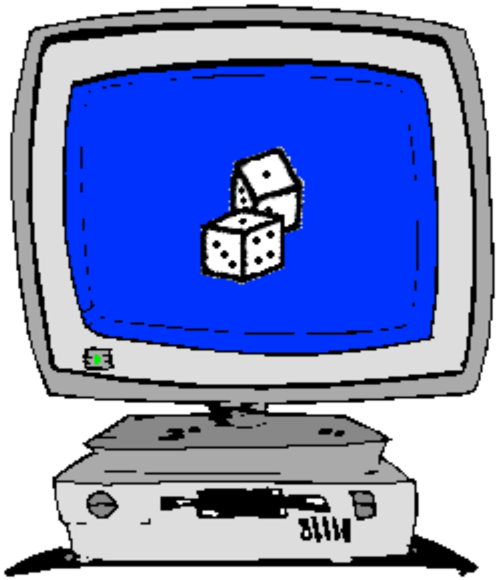
- *Metropolis* algorithm: model energy gap E as a probability

$$p \,\alpha\, e^{-E/kT}$$

- Integrals converge faster than any other method (for d > 3) when using *enough* independent events (central limit theorem)

- F. James, *Rep. Prog. Phys.*, Vol. **43** (1980) 1145.

Implementing Monte Carlo



Good random generator:

from thermal electronic noise (hardware)

or quasi-random generators => *quasi*-Monte-Carlo

We encounter a probability $0 < p < 1$.

Crude Monte-Carlo (yes/no choice):

We shoot n events $\xi \in [0,1]$

We keep events that satisfy $\xi < p$

np events $\rightarrow$ low statistics

Importance sampling (fuzzy choice – event weighting):

Keep n events, no more random number...

But associate a weight p to each of them (we set $\xi = p$)

Retain statistical accuracy

Software Packages

High Energy

- MCNP(X)
- PHITS
- FLUKA

Low Energy

- McSTAS
- Vitess
- RESTRAX
- NISP

GEANT

Calculation of radiation shielding and nuclear devices.

No coherent scattering

Calculation of neutron transport and scattering in neutron instruments.

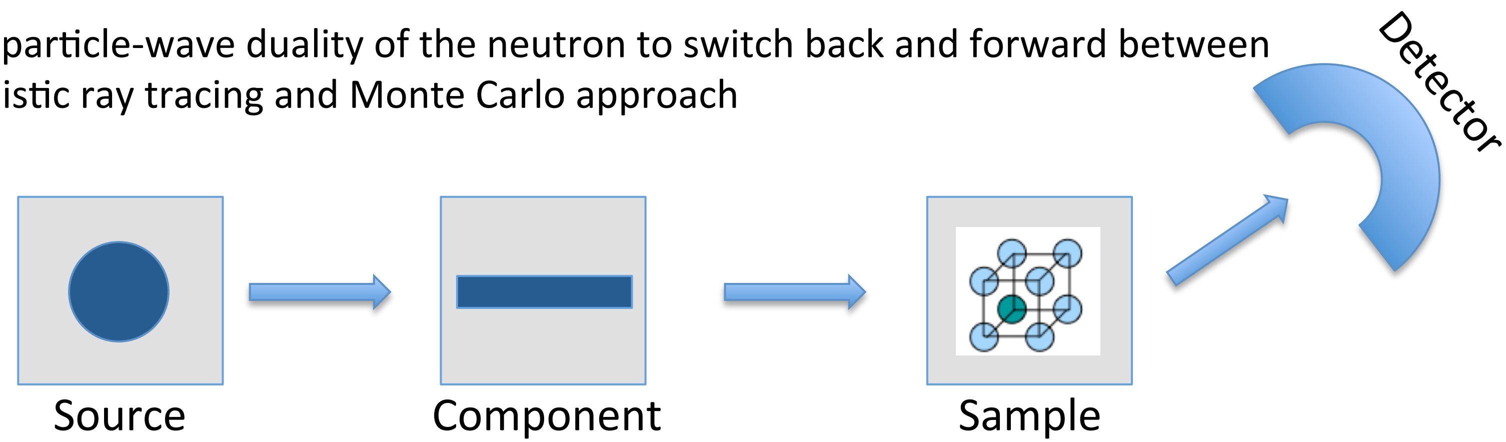
Includes coherent scattering

Simulating Neutron Transport and Scattering

Each time physics takes place (scattering, absorption, ...) random choices are made.

- Neutrons are described as **(r, v, s, t)**, and are transported along models.
- Deterministic propagation simply uses Newton rules, incl. gravitation.

- Instrument Monte Carlo methods implement coherent scattering effects
- Uses deterministic propagation where this can be done
- Uses Monte Carlo sampling of "complicated" distributions and stochastic processes and multiple outcomes with known probabilities are involved
  - I.e. inside scattering matter
- Uses the particle-wave duality of the neutron to switch back and forward between deterministic ray tracing and Monte Carlo approach

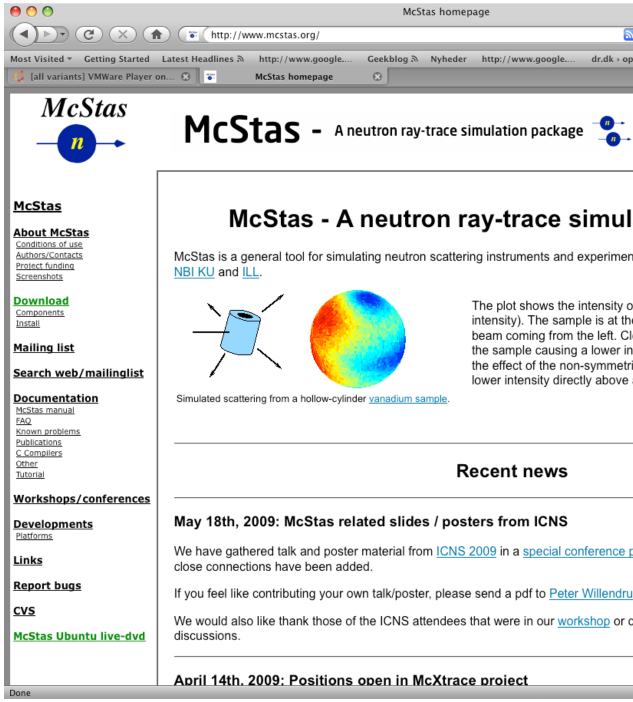


- Result: A realistic and efficient transport of neutrons in the thermal and cold range

McSTAS

GNU GPL
license
Open Source

- 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 2.5+1 people full time plus students

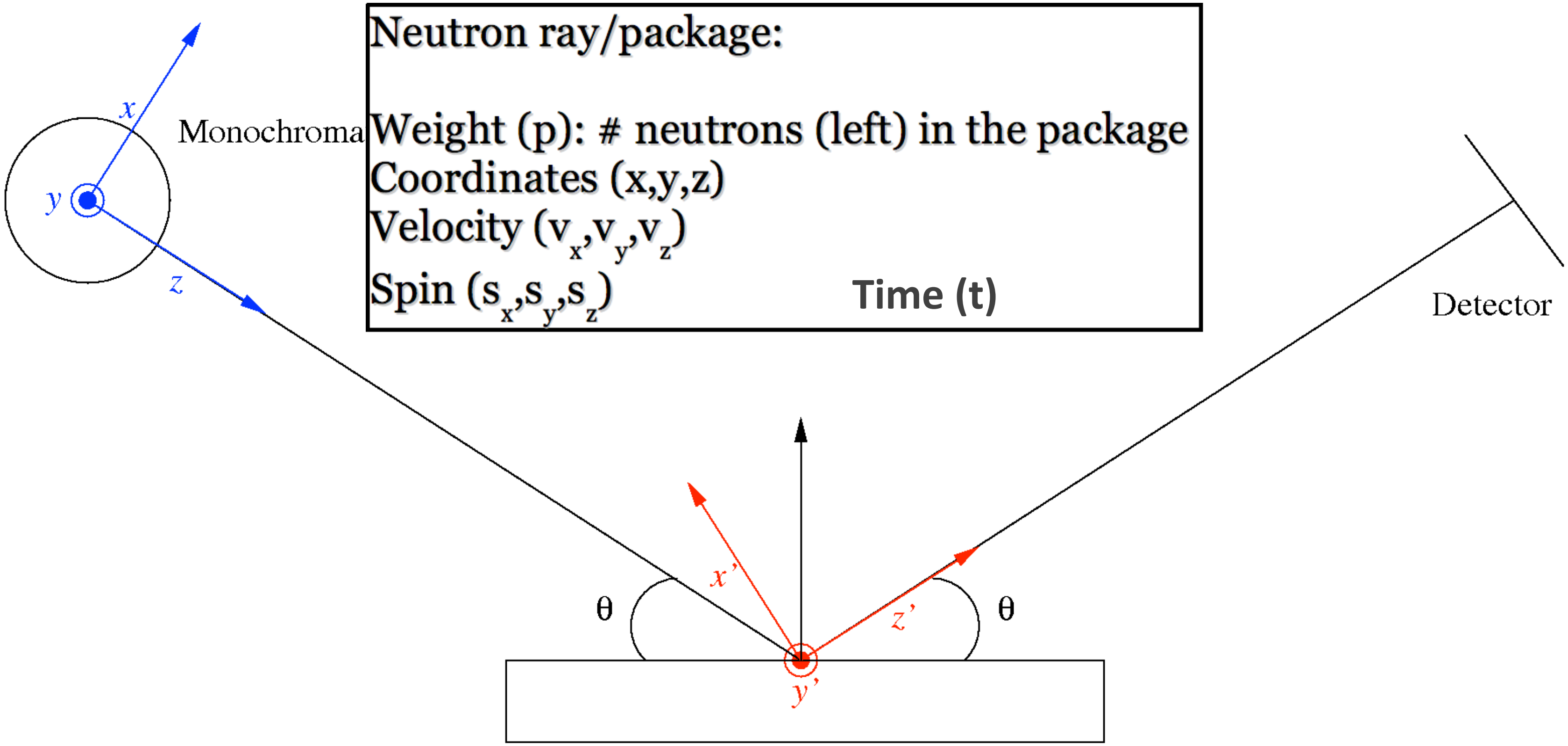


Project website at http://www.mcstas.org mcstas-users@mcstas.org mailinglist

How Does McSTAS Work?

Neutron ray/package:

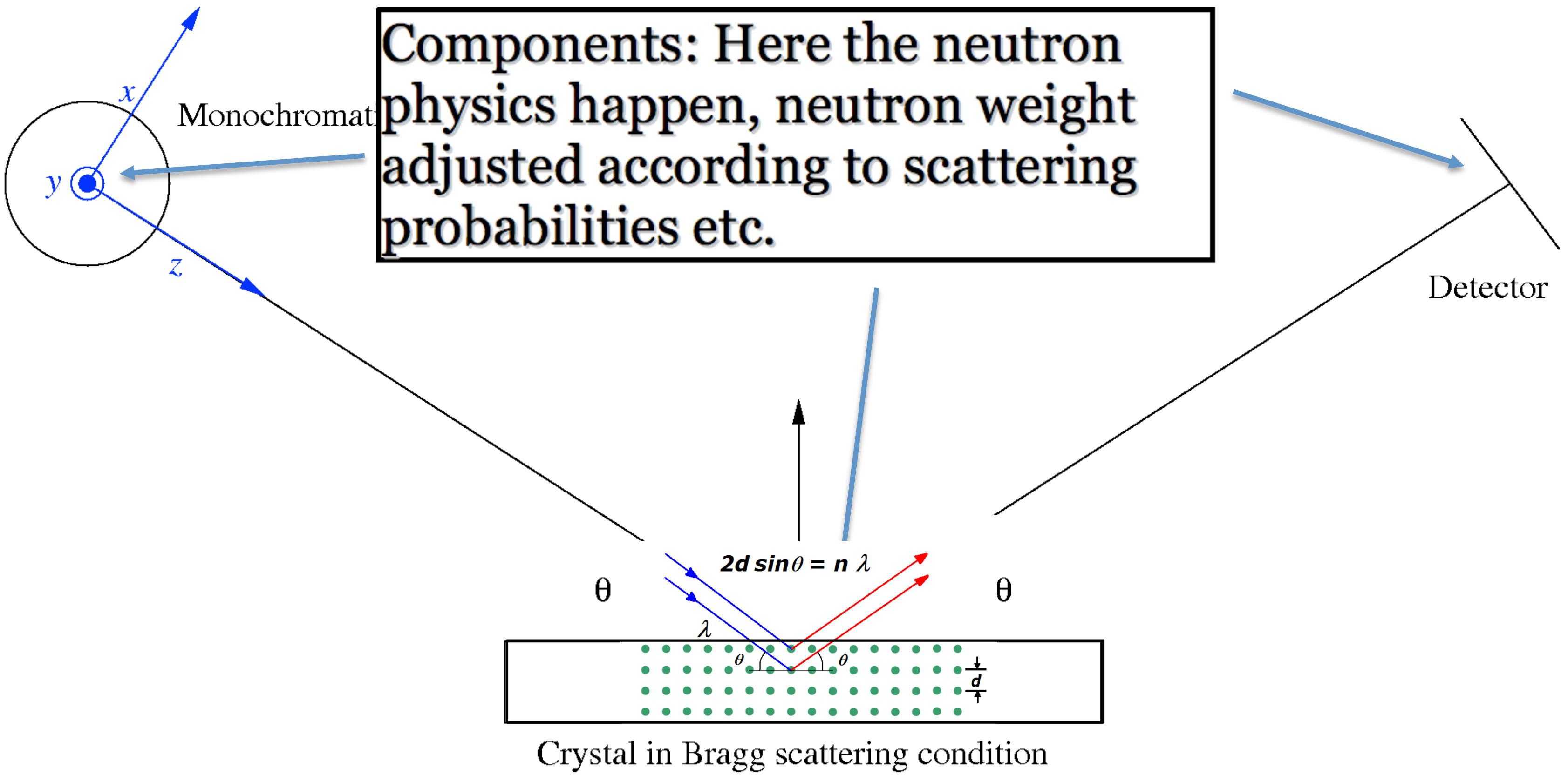
Weight (p): # neutrons (left) in the package
Coordinates (x,y,z)
Velocity (v_x,v_y,v_z)
Spin (s_x,s_y,s_z)
Time (t)



Crystal in Bragg scattering condition

How Does McSTAS Work?

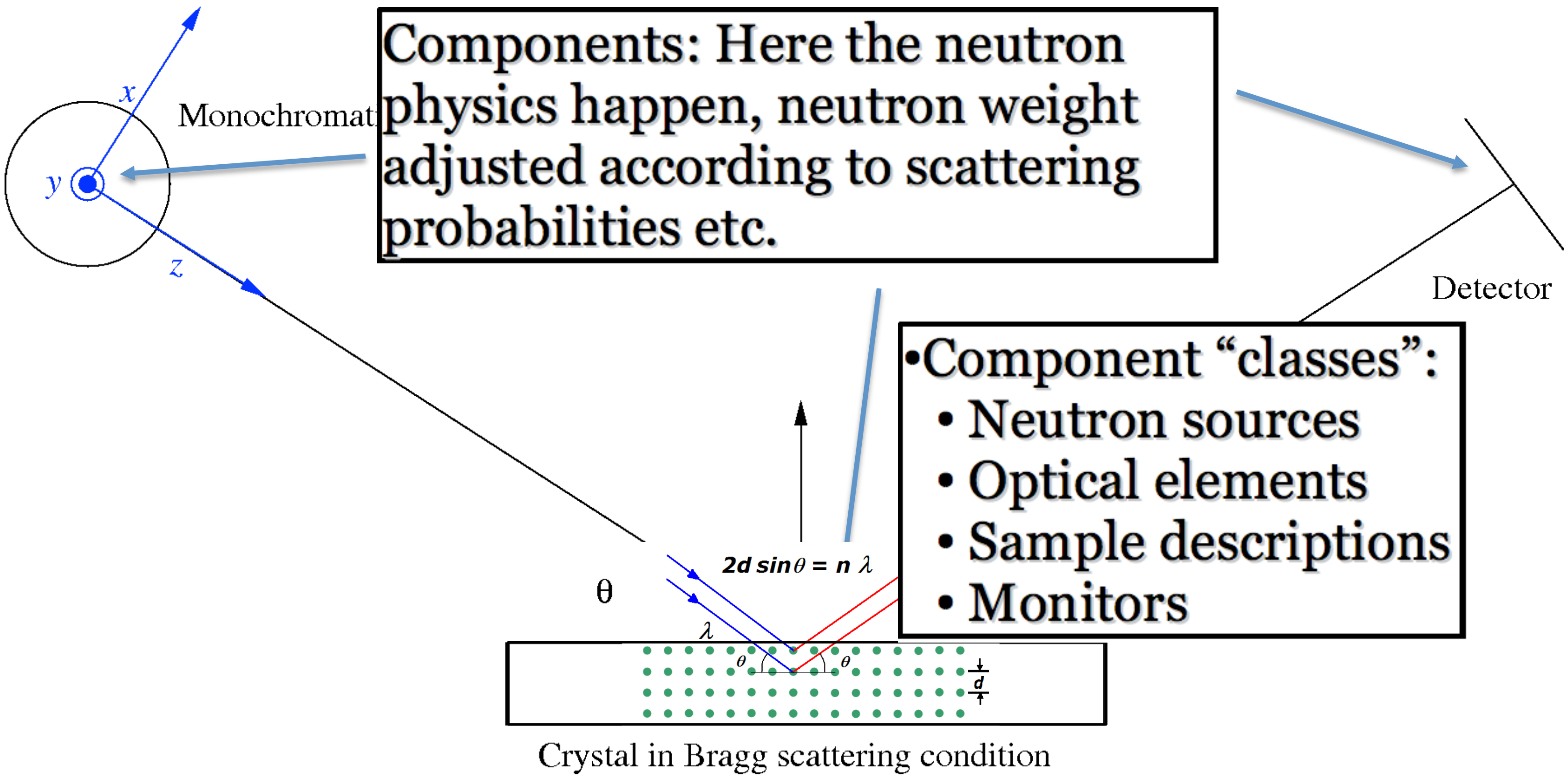
Components: Here the neutron physics happen, neutron weight adjusted according to scattering probabilities etc.



How Does McSTAS Work?

Components: Here the neutron physics happen, neutron weight adjusted according to scattering probabilities etc.

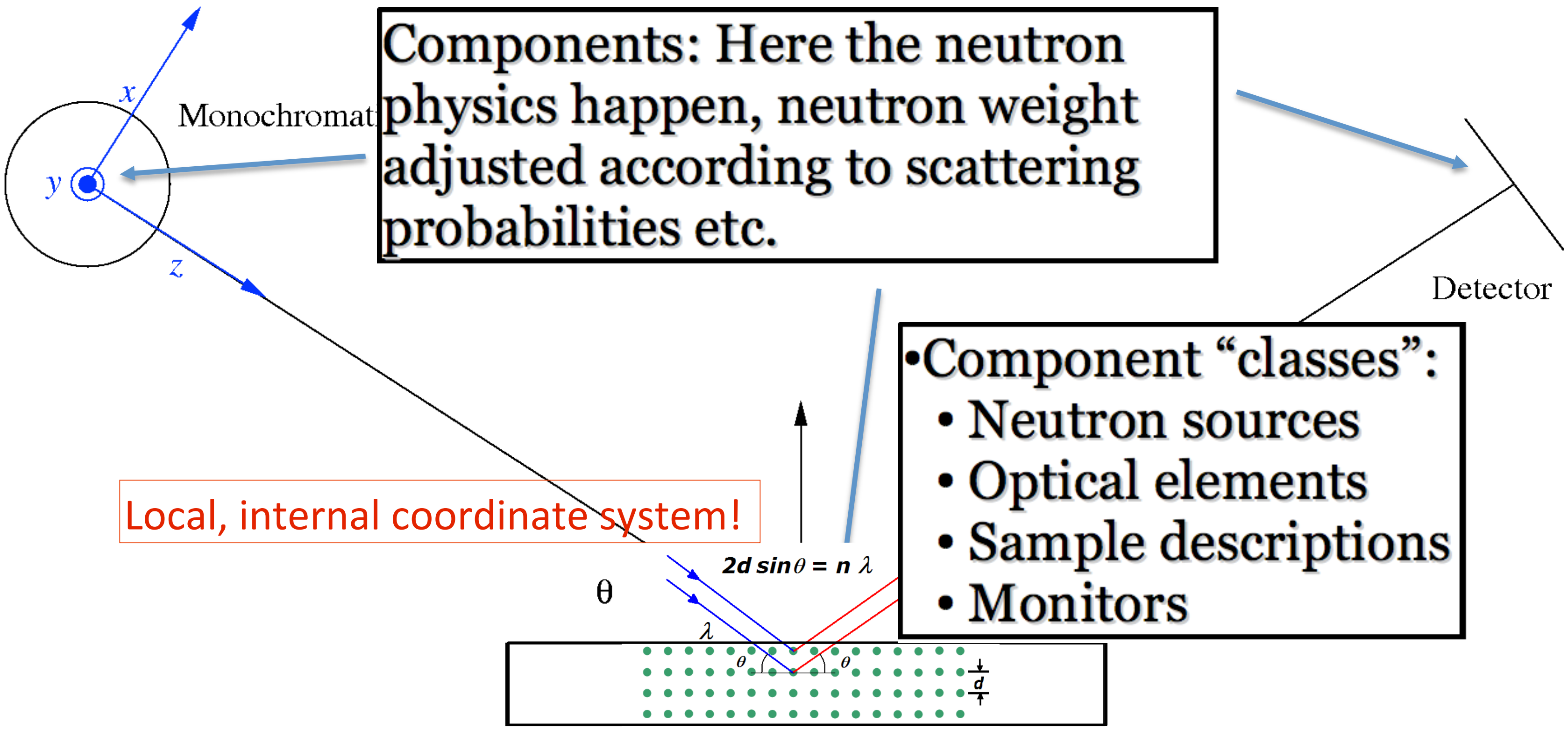
- Component "classes":
  - Neutron sources
  - Optical elements
  - Sample descriptions
  - Monitors



How Does McSTAS Work?

Components: Here the neutron physics happen, neutron weight adjusted according to scattering probabilities etc.

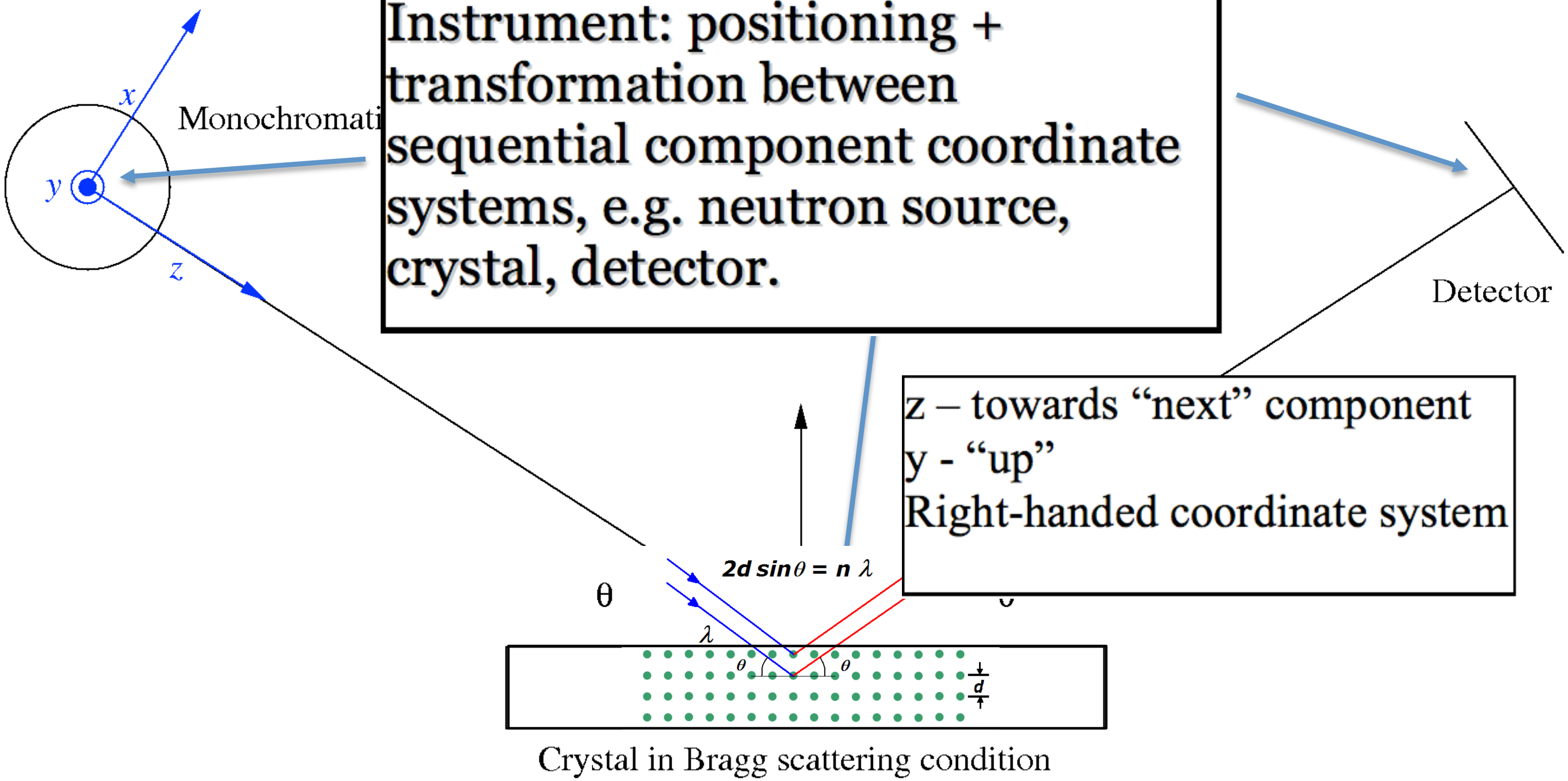
- Component "classes":
  - Neutron sources
  - Optical elements
  - Sample descriptions
  - Monitors



Local, internal coordinate system!

Crystal in Bragg scattering condition

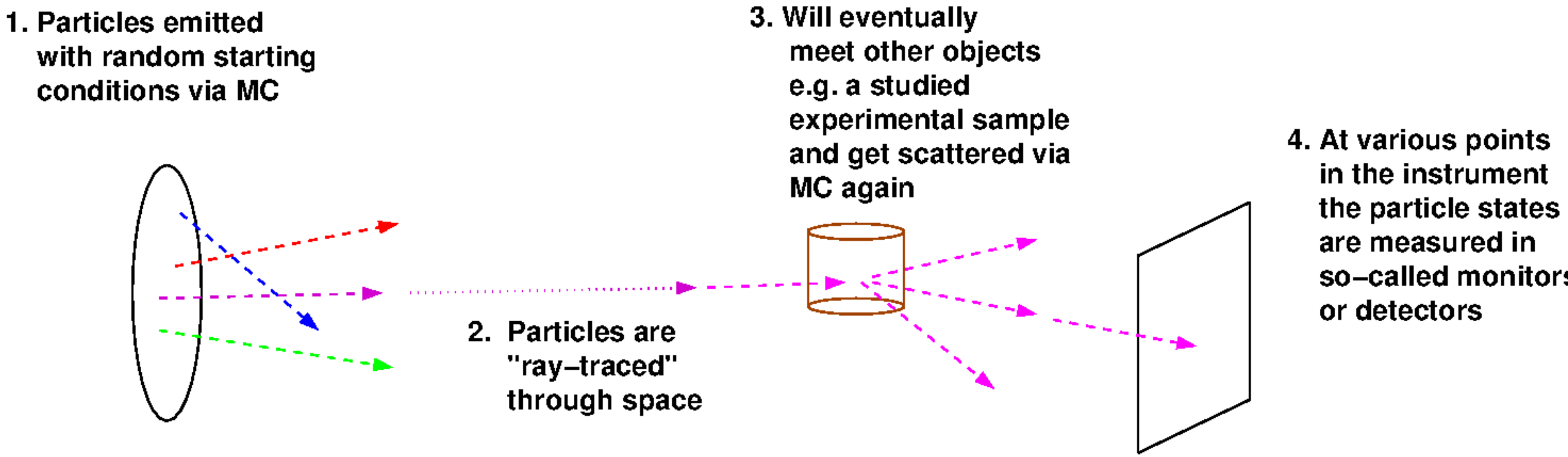
How Does McSTAS Work?



Instrument: positioning + transformation between sequential component coordinate systems, e.g. neutron source, crystal, detector.

z – towards "next" component
y - "up"
Right-handed coordinate system

How Does McSTAS Work?



Three levels of **source code** :

- **Instrument file** (All users) – Describes instrument and defines order and parameters of instrument components, samples etc.
- **Component files** (Some users) – Implements physics of component
- **ANSI C Code** (no users) – Common core routines, transport between components etc.

Instrument File

Written by User – You!

```
DEFINE INSTRUMENT My_Instrument(DIST=10)

/* Here comes the TRACE section, where the actual      */
/* instrument is defined as a sequence of components.  */
TRACE

/* The Arm() class component defines reference points and orientations  */
/* in 3D space.                                                         */
COMPONENT Origin = Arm()
  AT (0,0,0) ABSOLUTE


COMPONENT Source = Source_simple(
    radius = 0.1, dist = 10, xw = 0.1, yh = 0.1, E0 = 5, dE = 1)
  AT (0, 0, 0) RELATIVE Origin


COMPONENT Emon = E_monitor(
    filename = "Emon.dat", xmin = -0.1, xmax = 0.1, ymin = -0.1,
    ymax = 0.1, Emin = 0, Emax = 10)
  AT (0, 0, DIST) RELATIVE Origin


COMPONENT PSD = PSD_monitor(
    nx = 128, ny = 128, filename = "PSD.dat", xmin = -0.1,
    xmax = 0.1, ymin = -0.1, ymax = 0.1)
  AT (0, 0, 1e-10) RELATIVE Emon

/* The END token marks the instrument definition end */
END
```

Component File

Written by Developers or maybe by User

```
/*******************************************************************************
*
* Mcstas, neutron ray-tracing package
*         Copyright 1997-2002, All rights reserved
*         Risoe National Laboratory, Roskilde, Denmark
*         Institut Laue Langevin, Grenoble, France
*
* Component: Source_flat
*
* %I
* Written by: Kim Lefmann
* Date: October 30, 1997
* Modified by: KL, October 4, 2001
* Modified by: Emmanuel Farhi, October 30, 2001. Serious bug corrected.
* Version: $Revision: 1.22 $
* Origin: Risoe
* Release: McStas 1.6
*
* A circular neutron source with flat energy spectrum and arbitrary flux
*
* %D
* The routine is a circular neutron source, which aims at a square target
* centered at the beam (in order to improve MC-acceptance rate).  The angular
* divergence is then given by the dimensions of the target.
* The neutron energy is uniformly distributed between E0-dE and E0+dE.
*
* Example: Source_flat(radius=0.1, dist=2, xw=.1, yh=.1, E0=14, dE=2)
*
* %P
* radius: (m)   Radius of circle in (x,y,0) plane where neutrons
*               are generated.
* dist:   (m)   Distance to target along z axis.
* xw:     (m)   Width(x) of target
* yh:     (m)   Height(y) of target
* E0:     (meV) Mean energy of neutrons.
* dE:     (meV) Energy spread of neutrons.
* Lambda0 (AA)  Mean wavelength of neutrons.
* dLambda (AA)  Wavelength spread of neutrons.
* flux    (1/(s*cm**2*st)) Energy integrated flux
*
* %E
*******************************************************************************/

DEFINE COMPONENT Source_simple
DEFINITION PARAMETERS ()
SETTING PARAMETERS (radius, dist, xw, yh, E0=0, dE=0, Lambda0=0, dLambda=0, flux=1)
OUTPUT PARAMETERS ()
STATE PARAMETERS (x,y,z,vx,vy,vz,t,s1,s2,p)
DECLARE
%{
  double pmul, pdir;
%}
INITIALIZE
%{
  pmul=flux*PI*1e4*radius*radius/mcget_ncount();
%}
TRACE
%{
 double chi,E,Lambda,v,r, xf, yf, rf, dx, dy;

 t=0;
 z=0;

 chi=2*PI*rand01();                          /* Choose point on source */
 r=sqrt(rand01())*radius;                    /* with uniform distribution. */
 x=r*cos(chi);
 y=r*sin(chi);

 randvec_target_rect(&xf, &yf, &rf, &pdir,
        0, 0, dist, xw, yh, ROT_A_CURRENT_COMP);

 dx = xf-x;
 dy = yf-y;
 rf = sqrt(dx*dx+dy*dy+dist*dist);

 p = pdir*pmul;

 if(Lambda0==0) {
   E=E0+dE*randpm1();              /*  Choose from uniform distribution */
   v=sqrt(E)*SE2V;
 } else {
   Lambda=Lambda0+dLambda*randpm1();
   v = K2V*(2*PI/Lambda);
 }

 vz=v*dist/rf;
 vy=v*dy/rf;
 vx=v*dx/rf;
%}

MCDISPLAY
%{
  magnify("xy");
  circle("xy",0,0,0,radius);
%}

END
```

Generated C-Code

Written by McStas

```
/* Automatically generated file. Do not edit.
 * Format:     ANSI C source code
 * Creator:    McStas <http://neutron.risoe.dk>
 * Instrument: My_Instrument.instr (My_Instrument)
 * Date:       Sat Apr  9 15:27:56 2005
 */


/* THOUSANDS of lines removed here.... */

  /* TRACE Component Source. */
  SIG_MESSAGE("Source (Trace)");
  mcDEBUG_COMP("Source")
  mccoordschange(mcposrSource, mcrotrSource,
    &mcnlx, &mcnly, &mcnlz,
    &mcnlvx, &mcnlvy, &mcnlvz,
    &mcnlt, &mcnlsx, &mcnlsy);
  mcDEBUG_STATE(mcnlx, mcnly, mcnlz, mcnlvx, mcnlvy, mcnlvz,mcnlt,mcnlsx,mcnlsy, mcnlp)
#define x mcnlx
#define y mcnly
#define z mcnlz
#define vx mcnlvx
#define vy mcnlvy
#define vz mcnlvz
#define t mcnlt
#define s1 mcnlsx
#define s2 mcnlsy
#define p mcnlp
  STORE_NEUTRON(2,mcnlx, mcnly, mcnlz, mcnlvx,mcnlvy,mcnlvz,mcnlt,mcnlsx,mcnlsy, mcnlsz, mcnlp);
  mcScattered=0;
  mcNCounter[2]++;
#define mccompcurname Source
#define mccompcurindex 2
{   /* Declarations of SETTING parameters. */
MCNUM radius = mccSource_radius;
MCNUM dist = mccSource_dist;
MCNUM xw = mccSource_xw;
MCNUM yh = mccSource_yh;
MCNUM E0 = mccSource_E0;
MCNUM dE = mccSource_dE;
MCNUM Lambda0 = mccSource_Lambda0;
MCNUM dLambda = mccSource_dLambda;
MCNUM flux = mccSource_flux;
#line 58 "Source_simple.comp"
{
 double chi,E,Lambda,v,r, xf, yf, rf, dx, dy;
```

McStas is a (pre)compiler

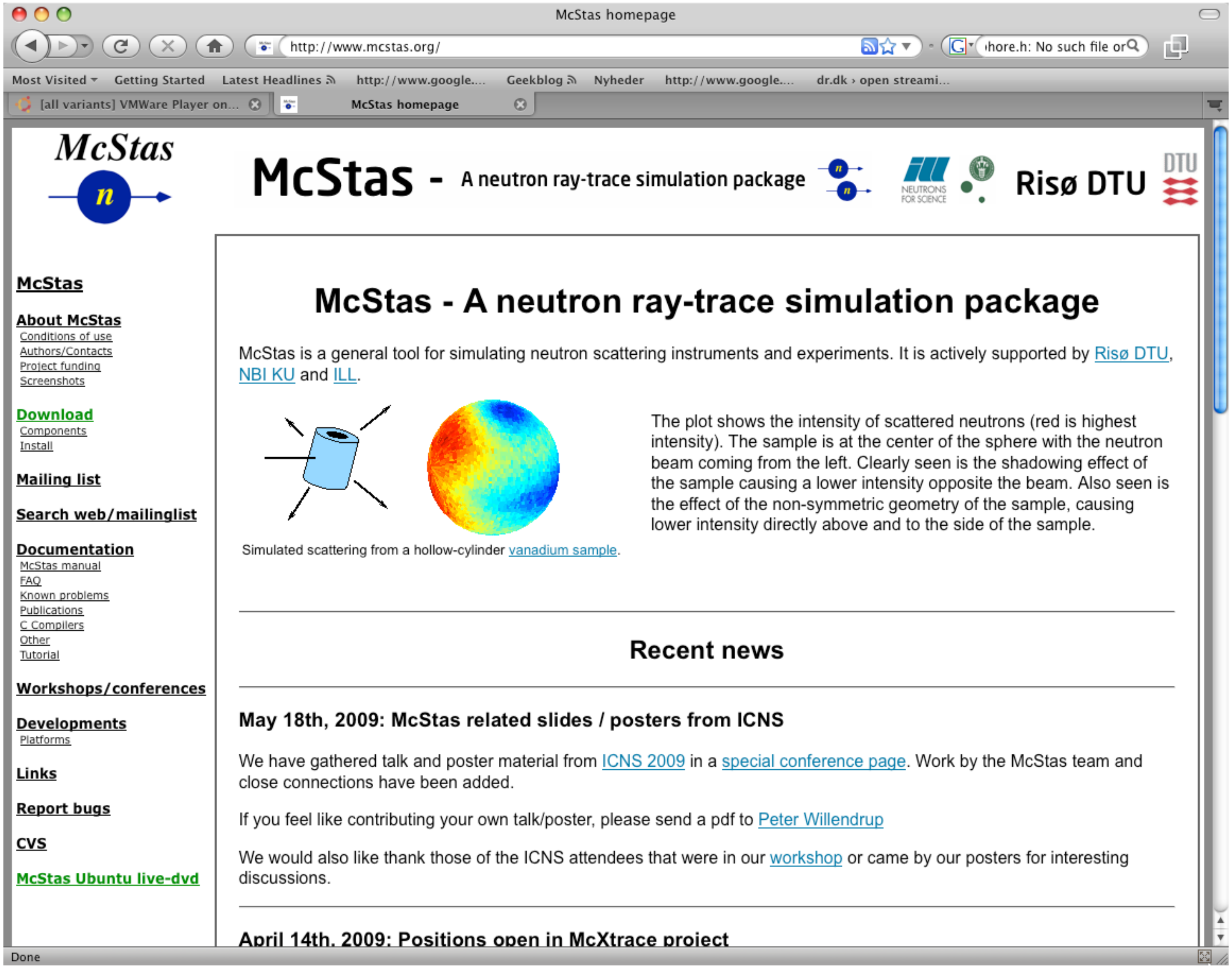
Input is .instr and .comp files + runtime functions from core library for e.g. random numbers.

Output is a single c-file, which can be compiled with your favourite C compiler.

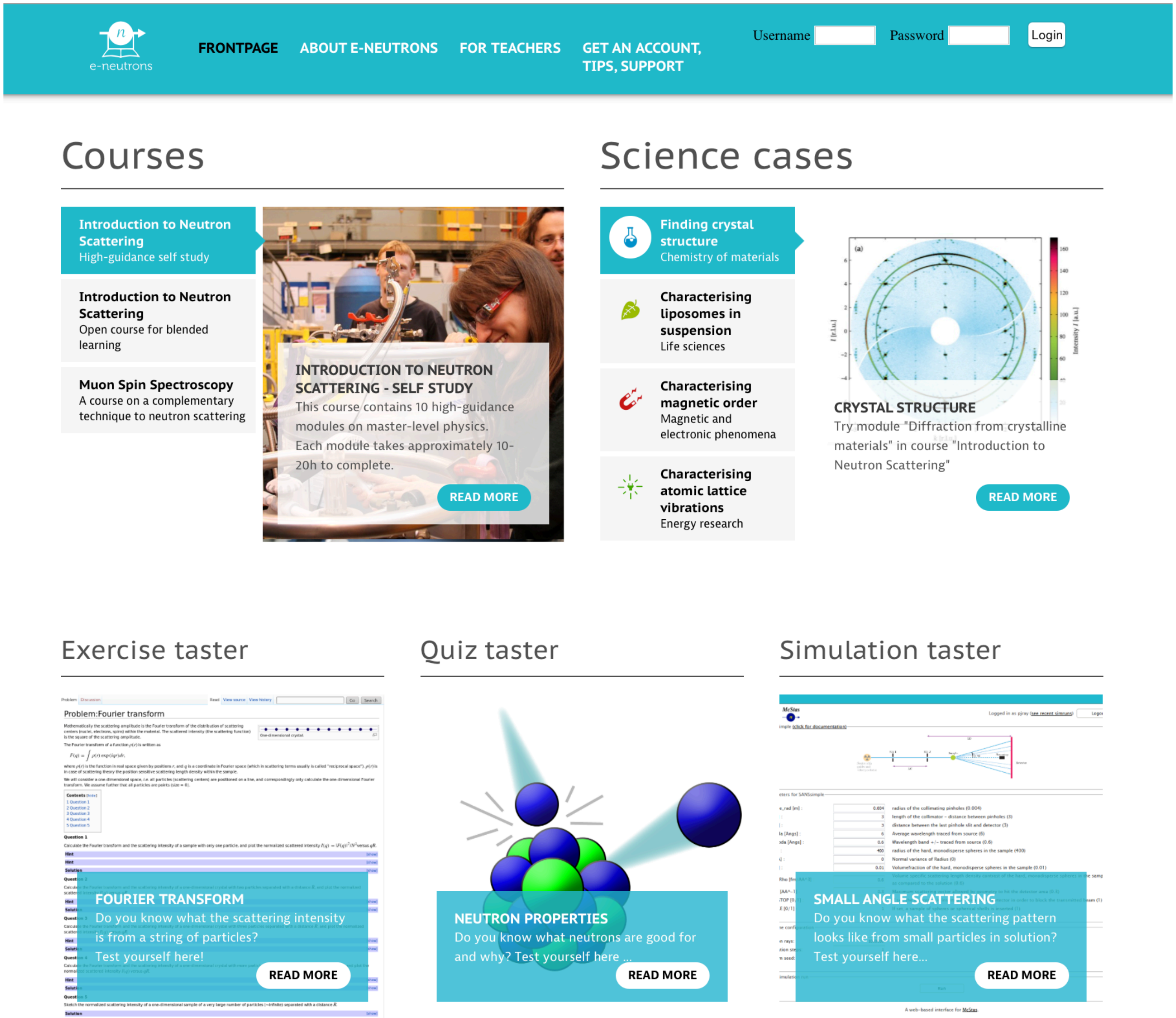
Can be made to take input arguments if required, for e.g. scans of instrument parameters

Live Demo

Try it out!


McStas homepage
http://www.mcstas.org/
Most Visited
Getting Started
Latest Headlines
Geekblog
Nyheder
dr.dk › open streami...
[all variants] VMWare Player on...
McStas homepage
McStas
McStas - A neutron ray-trace simulation package
Risø DTU
McStas
About McStas
Conditions of use
Authors/Contacts
Project funding
Screenshots
Download
Components
Install
Mailing list
Search web/mailinglist
Documentation
McStas manual
FAQ
Known problems
Publications
C Compilers
Other
Tutorial
Workshops/conferences
Developments
Platforms
Links
Report bugs
CVS
McStas Ubuntu live-dvd
McStas - A neutron ray-trace simulation package
McStas is a general tool for simulating neutron scattering instruments and experiments. It is actively supported by Risø DTU, NBI KU and ILL.
The plot shows the intensity of scattered neutrons (red is highest intensity). The sample is at the center of the sphere with the neutron beam coming from the left. Clearly seen is the shadowing effect of the sample causing a lower intensity opposite the beam. Also seen is the effect of the non-symmetric geometry of the sample, causing lower intensity directly above and to the side of the sample.
Simulated scattering from a hollow-cylinder vanadium sample.
Recent news
May 18th, 2009: McStas related slides / posters from ICNS
We have gathered talk and poster material from ICNS 2009 in a special conference page. Work by the McStas team and close connections have been added.
If you feel like contributing your own talk/poster, please send a pdf to Peter Willendrup
We would also like thank those of the ICNS attendees that were in our workshop or came by our posters for interesting discussions.
April 14th, 2009: Positions open in McXtrace project
Done

Try it out!


e-neutrons
FRONTPAGE
ABOUT E-NEUTRONS
FOR TEACHERS
GET AN ACCOUNT, TIPS, SUPPORT
Username
Password
Login
Courses
Introduction to Neutron Scattering
High-guidance self study
Introduction to Neutron Scattering
Open course for blended learning
Muon Spin Spectroscopy
A course on a complementary technique to neutron scattering
INTRODUCTION TO NEUTRON SCATTERING - SELF STUDY
This course contains 10 high-guidance modules on master-level physics. Each module takes approximately 10-20h to complete.
READ MORE
Science cases
Finding crystal structure
Chemistry of materials
Characterising liposomes in suspension
Life sciences
Characterising magnetic order
Magnetic and electronic phenomena
Characterising atomic lattice vibrations
Energy research
CRYSTAL STRUCTURE
Try module "Diffraction from crystalline materials" in course "Introduction to Neutron Scattering"
READ MORE
Exercise taster
Problem:Fourier transform
FOURIER TRANSFORM
Do you know what the scattering intensity is from a string of particles?
Test yourself here!
READ MORE
Quiz taster
NEUTRON PROPERTIES
Do you know what neutrons are good for and why? Test yourself here ...
READ MORE
Simulation taster
SMALL ANGLE SCATTERING
Do you know what the scattering pattern looks like from small particles in solution?
Test yourself here...
READ MORE