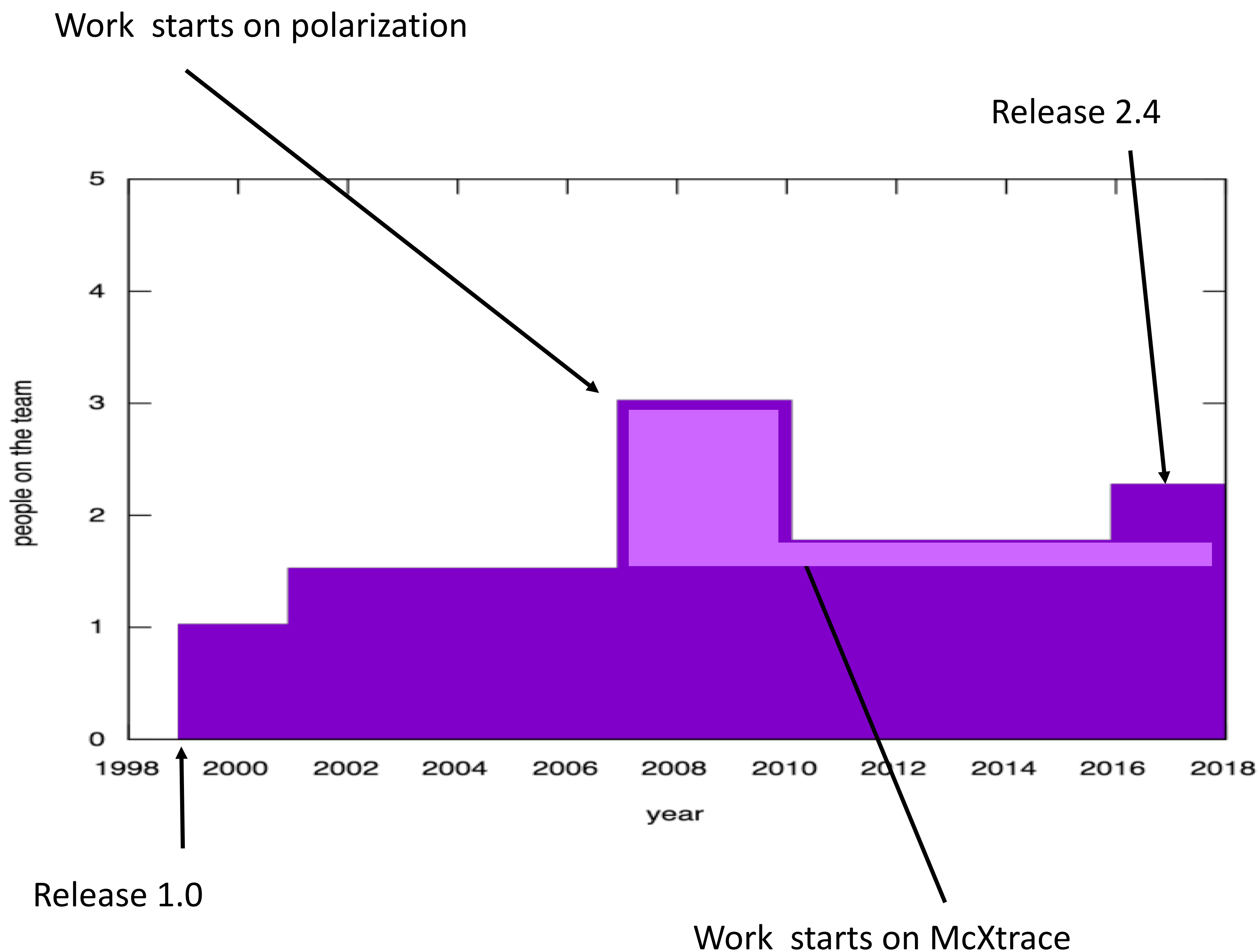
McStas

Simulating Polarized Neutron Scattering Experiments and Equipment with McStas

Emmanuel Farhi, ILL
(Slides from Erik Bergbäck Knudsen, DTU Physics)

Workforce



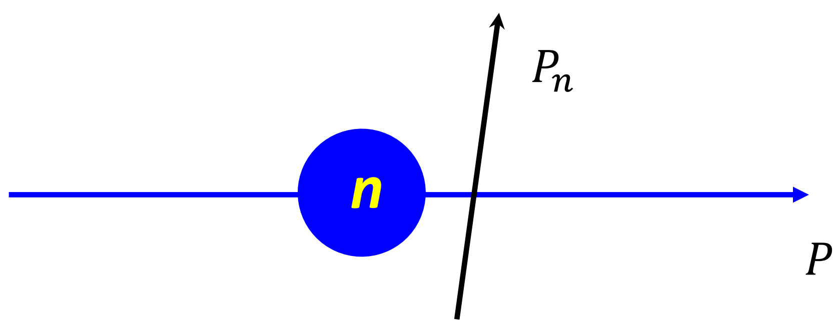
Work starts on polarization
Release 2.4
people on the team
0
1
2
3
4
5
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
year
Release 1.0
Work starts on McXtrace

Mcstas "particle" model

Neutron ray/package:

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

$$P_n = \frac{1}{p}\sum_{i}^{p} P_i \, , n = raynumber$$



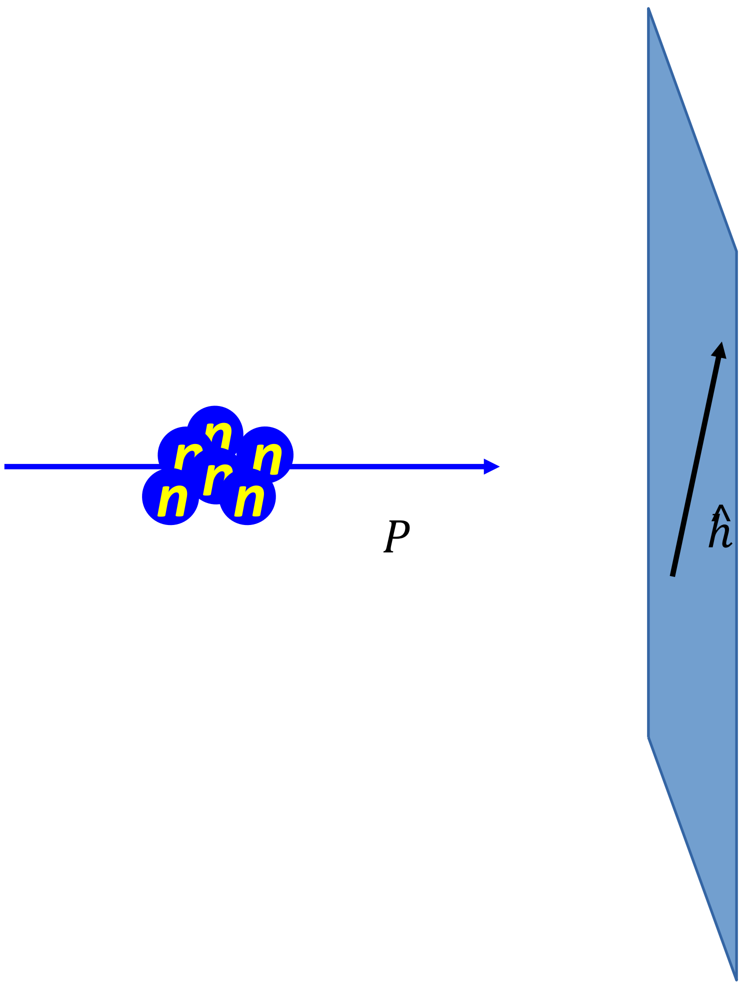
$$P_i = 2\left(\langle \hat{s}_{x,i}\rangle \hat{\imath}_{x,i} + \langle \hat{s}_{y,i}\rangle \hat{\imath}_{y,i} + \langle \hat{s}_{z,i}\rangle \hat{\imath}_{z,i}\right)$$

$$P = \frac{1}{N}\sum_{n=0}^{N} P_n$$

From G. Williams: *"Polarized neutrons", Oxford Science Publ.*, 1988

McStas detectors/monitors

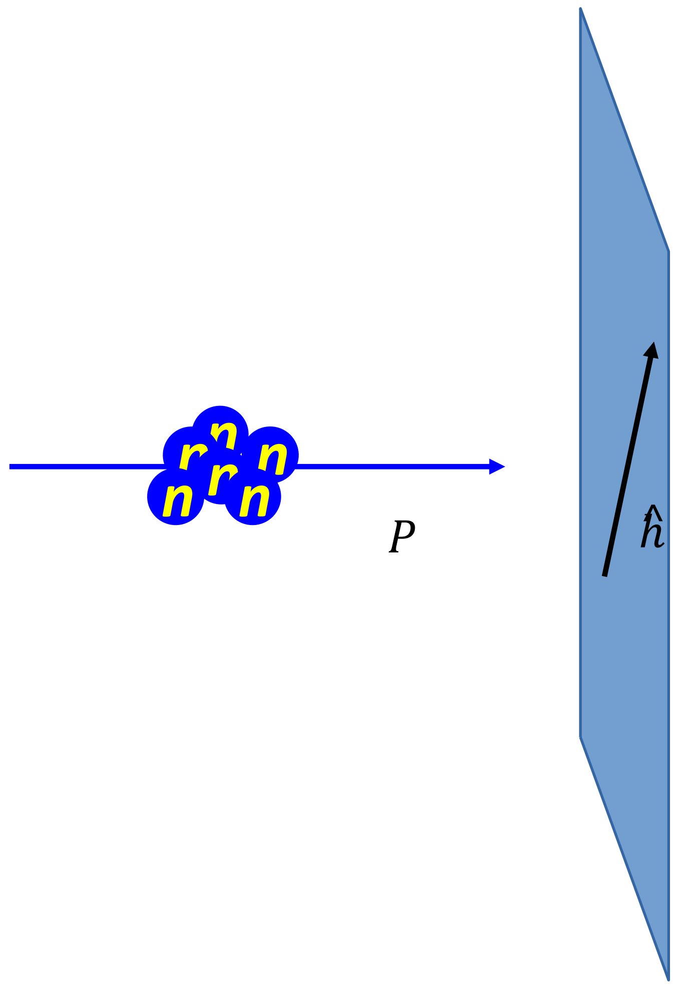
Monitoring: How and What do we monitor?

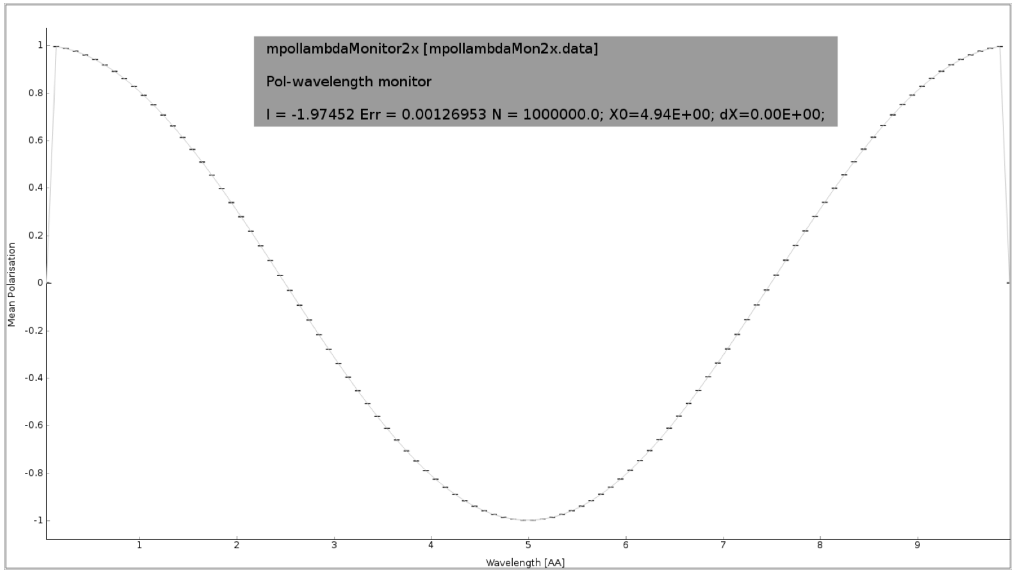


$$P_{\hat{h}} = \frac{\sum_{n=0}^{N} p_n \, P_n \cdot \hat{h}}{\sum_{n}^{N} p_n}$$

McStas detectors/monitors

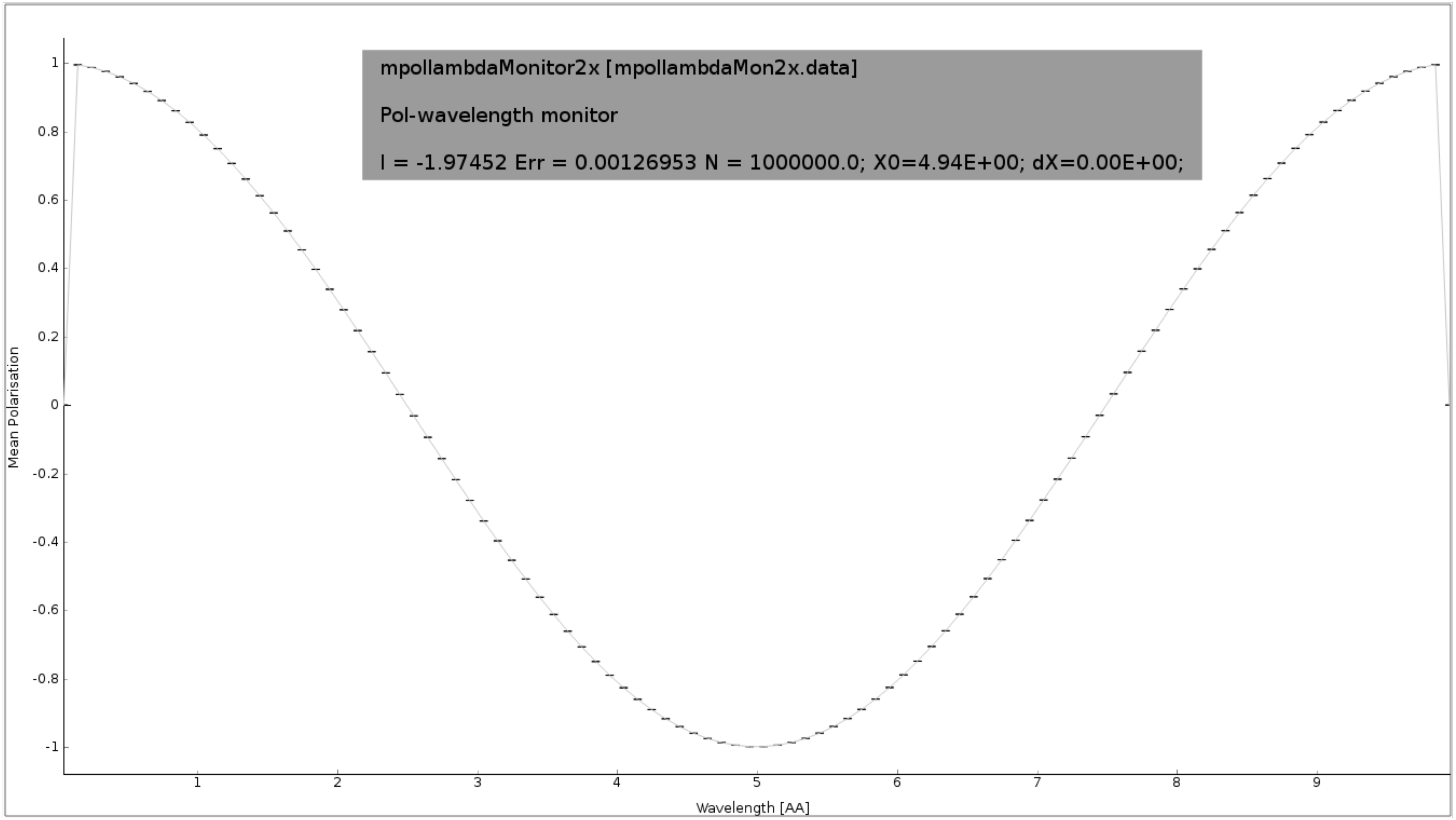
Monitoring: How and What do we monitor?





McStas detectors/monitors

Monitoring: How and What do we monitor?



Polarization monitors

- Available monitors:
- `Pol_monitor.comp`: 0D
- `PolLambda_monitor.comp`: 2D
- `MeanPolLambda_monitor.comp`: 1D

- Magnetic fields in McStas
- The challenge:
  - Fast beam/ray transport: # $rays > 10^6$
  - Unknown magnetic field and field strength
  - >1 Magnet → nested fields.

McStas precession algorithm

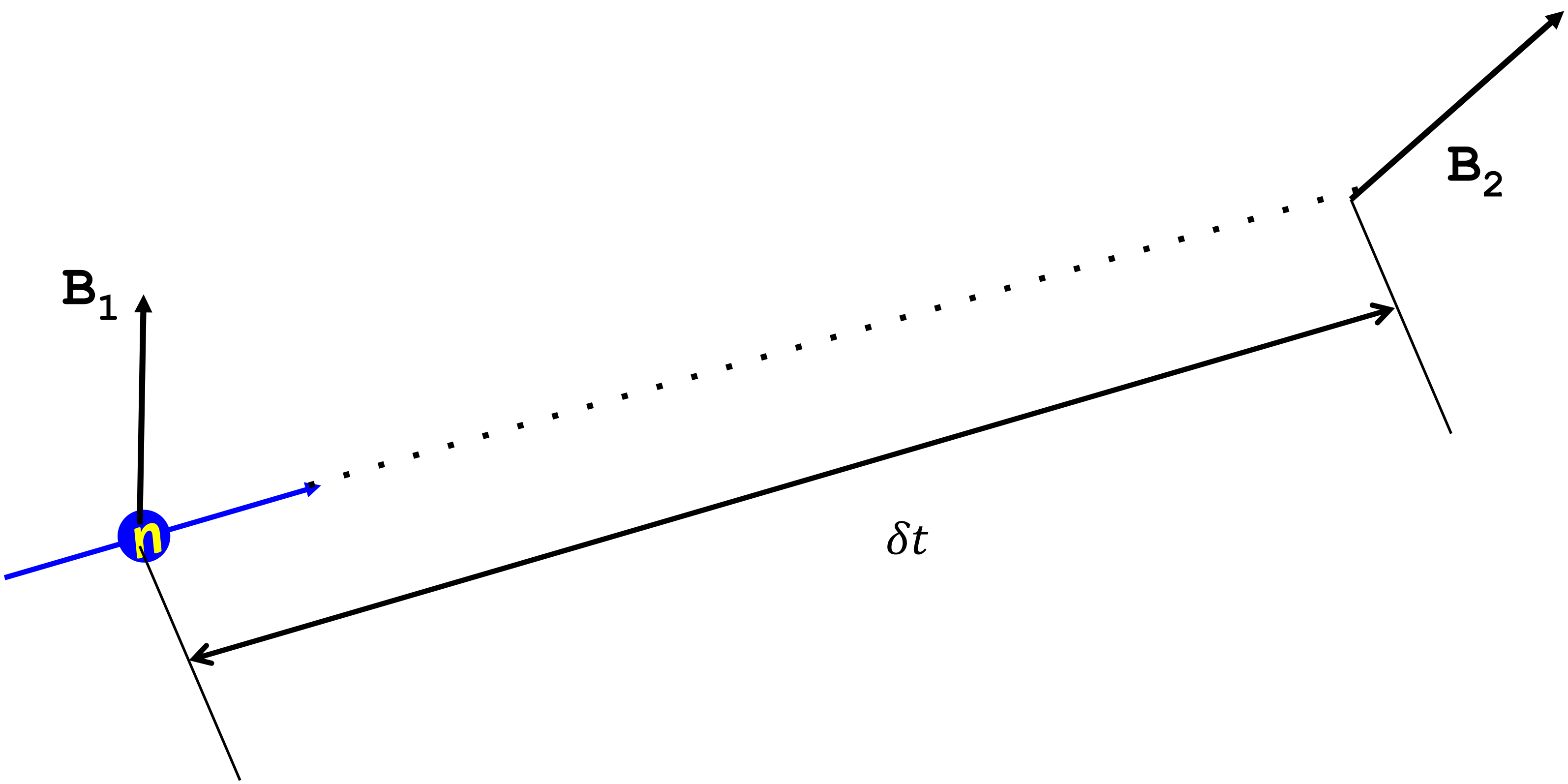
while $n_t < t_{target}$ **do**

- store neutron;
- sample magnetic field: $\mathbf{B}_1 = \mathbf{B}(n_x, n_y, n_z, n_t)$;
- propagate neutron: $\delta t (< \Delta t)$;
- sample magnetic field: $\mathbf{B}_2 = \mathbf{B}(n_x, n_y, n_z, n_t)$;
- **while** $|\mathbf{B}_1 - \mathbf{B}_2| > \delta B_{threshold}$ **do**
  - restore neutron;
  - $\delta t := \delta t / 2$;
  - propagate neutron: $\delta t (< \Delta t)$;
  - sample magnetic field: $\mathbf{B}_1 = \mathbf{B}(n_x, n_y, n_z, n_t)$;
- precess polarization: $\mathbf{P_n}$ by ω around $\frac{\mathbf{B}_1 + \mathbf{B}_2}{2}$;

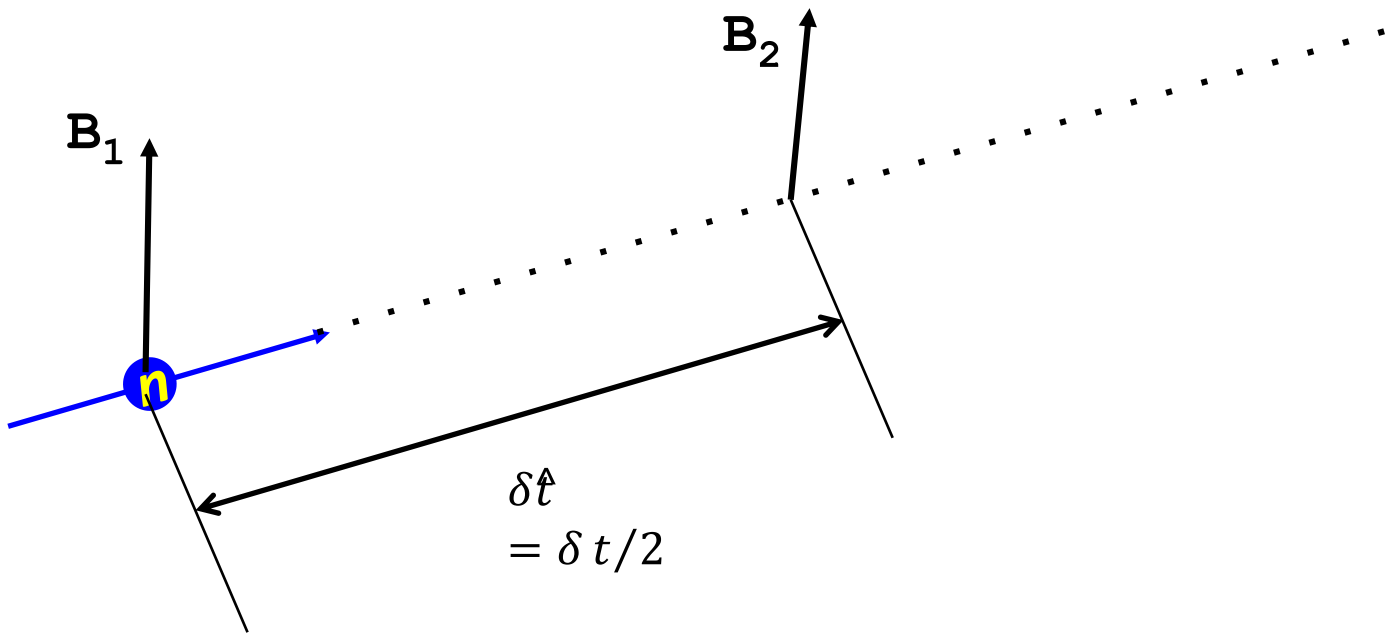
Algorithm 1: SimpleNumMagnetPrecession: Simplistic algorithm for tracking polarization of a Monte-Carlo neutron in a magnetic field. The neutron's state is stored as a position (n_x, n_y, n_z), a velocity $\mathbf{v}$, time n_t, and polarization vector $\mathbf{P_n}$.

From: Knudsen et.al., *J. Neutron Research*, 2014

McStas precession algorithm



McStas precession algorithm



McStas precession algorithm

while $n_t < t_{target}$ **do**

store neutron;

sample magnetic field: $\mathbf{B}_1 = \mathbf{B}(n_x, n_y, n_z, n_t)$;

propagate neutron: $\delta t (< \Delta t)$;

sample magnetic field: $\mathbf{B}_2 = \mathbf{B}(n_x, n_y, n_z, n_t)$;

while $|\mathbf{B}_1 - \mathbf{B}_2| > \delta B_{threshold}$ **do**

restore neutron;

$\delta t := \delta t / 2$;

propagate neutron: $\delta t (< \Delta t)$;

sample magnetic field: $\mathbf{B}_1 = \mathbf{B}(n_x, n_y, n_z, n_t)$;

precess polarization: $\mathbf{P_n}$ by ω around $\frac{\mathbf{B}_1 + \mathbf{B}_2}{2}$;

Algorithm 1: SimpleNumMagnetPrecession: Simplistic algorithm for tracking polarization of a Monte-Carlo neutron in a magnetic field. The neutron's state is stored as a position (n_x, n_y, n_z), a velocity $\mathbf{v}$, time n_t, and polarization vector $\mathbf{P_n}$.

From: Knudsen et.al., *J. Neutron Research*, 2014

McStas precession algorithm

while $n_t < t_{target}$ **do**

- store neutron;
- sample magnetic field: $\mathbf{B}_1 = \mathbf{B}(n_x, n_y, n_z, n_t)$;
- propagate neutron: $\delta t(< \Delta t)$;
- sample magnetic field: $\mathbf{B}_2 = \mathbf{B}(n_x, n_y, n_z, n_t)$;
- **while** $|\mathbf{B}_1 - \mathbf{B}_2| > \delta B_{threshold}$ **do**
  - restore neutron;
  - $\delta t := \delta t/2$;
  - propagate neutron: $\delta t(< \Delta t)$;
  - sample magnetic field: $\mathbf{B}_1 = \mathbf{B}(n_x, n_y, n_z, n_t)$;
- precess polarization: $\mathbf{P_n}$ by ω around $\frac{\mathbf{B}_1 + \mathbf{B}_2}{2}$;

Algorithm 1: SimpleNumMagnetPrecession: Simplistic algorithm for tracking polarization of a Monte-Carlo neutron in a magnetic field. The neutron's state is stored as a position (n_x, n_y, n_z), a velocity $\mathbf{v}$, time n_t, and polarization vector $\mathbf{P_n}$.

From: Knudsen et.al., *J. Neutron Research*, 2014

McStas precession algorithm

while $n_t < t_{target}$ **do**

store neutron;

sample magnetic f eld

```
void mc_pol_set_timestep(double dt);
```

propagate neutron: $\delta t(< \Delta t)$;

sample magnet

```
void mc_pol_set_angular_accuracy(double domega);
```

while $|\mathbf{B}_1 - \mathbf{B}_2| > \delta B_{threshold}$ **do**

restore neutron;

$\delta t := \delta t/2$;

propagate neutron: $\delta t(< \Delta t)$;

sample magnetic f eld: $\mathbf{B}_1 = \mathbf{B}(n_x, n_y, n_z, n_t)$;

precess polarization: $\mathbf{P_n}$ by ω around $\frac{\mathbf{B}_1 + \mathbf{B}_2}{2}$;

Algorithm 1: SimpleNumMagnetPrecession: Simplistic algorithm for tracking polarization of a Monte-Carlo neutron in a magnetic f eld. The neutron's state is stored as a position (n_x, n_y, n_z), a velocity $\mathbf{v}$, time n_t, and polarization vector $\mathbf{P_n}$.

From: Knudsen et.al., *J. Neutron Research*, 2014

McStas polarization components

Magnetic fields:
- `Pol_FieldBox.comp`
- `Pol_constBfield.comp`
- `Pol_simpleBfield.comp`
- `Pol_simpleBfield_stop.comp`
- `Pol_triafield.comp`

Optics:
- `Monochromator_pol.comp`
- `Pol_bender.comp`
- `Pol_guide_vmirror.comp`
- `Pol_mirror.comp`
- `Pol_pi_2_rotator.comp`
- `Transmission_polarisatorABSnT.comp`
- `Pol_bender_tapering.comp`

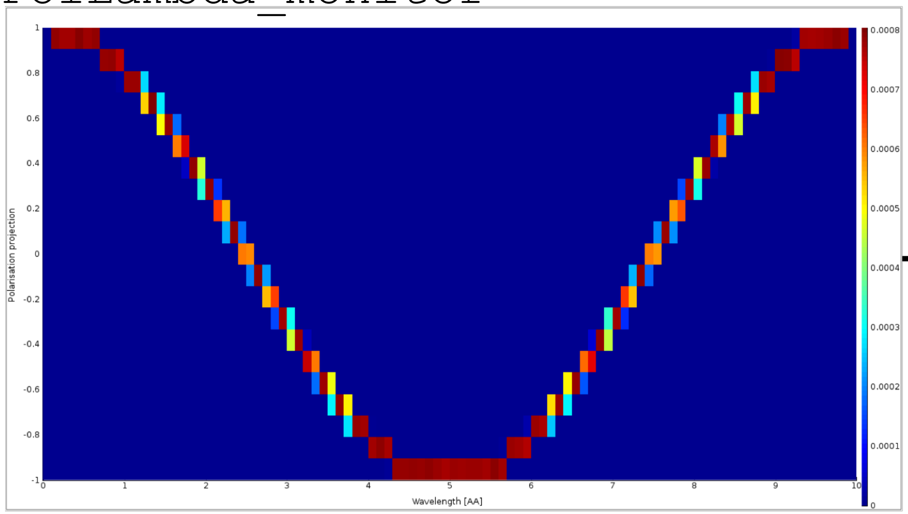
Monitors:
- `Pol_monitor.comp`
- `MeanPolLambda_monitor.comp`
- `PolLambda_monitor.comp`

Idealized components:
- `PolAnalyser_ideal.comp`
- `Set_pol.comp`

Contrib:
- `Foil_flipper_magnet.comp`

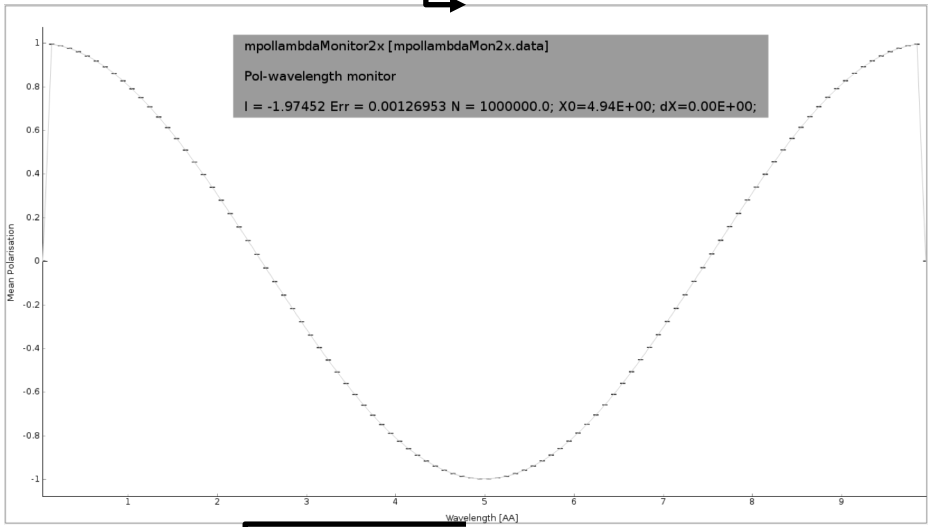
McStas polarization monitors

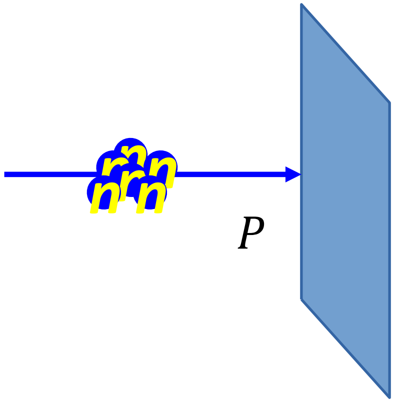
PolLambda_monitor



Monitors

MeanPolLambda_monitor



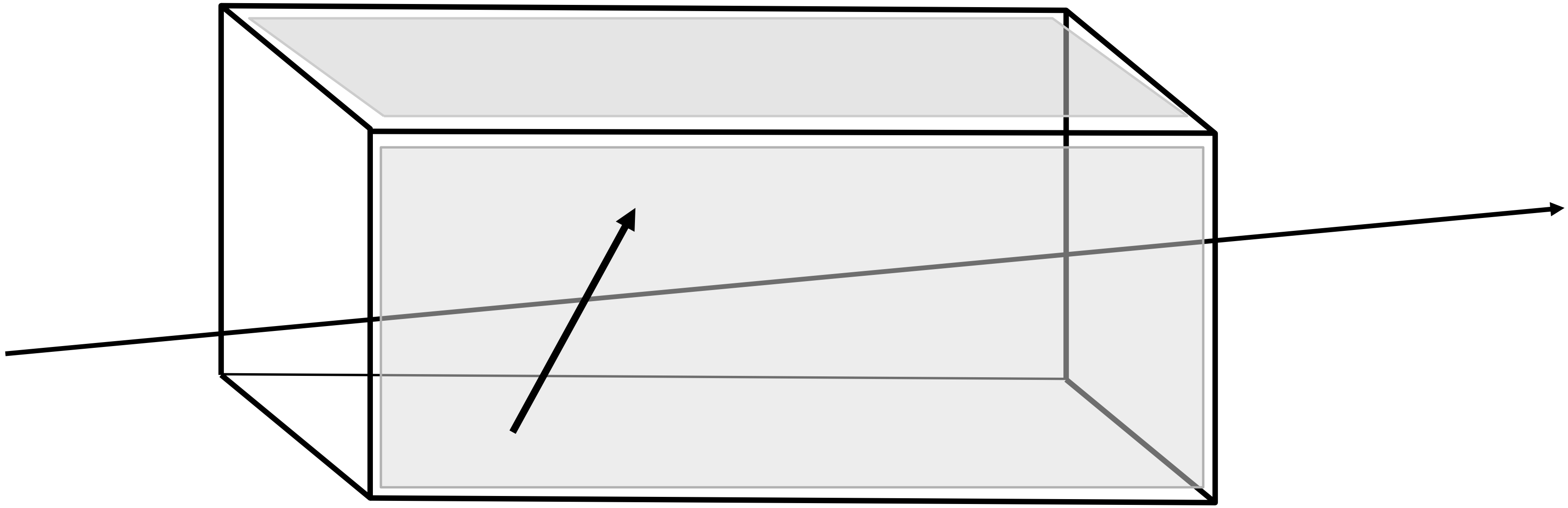


Pol_monitor

$$P \parallel (m_x, m_y, m_z) = 0.87$$

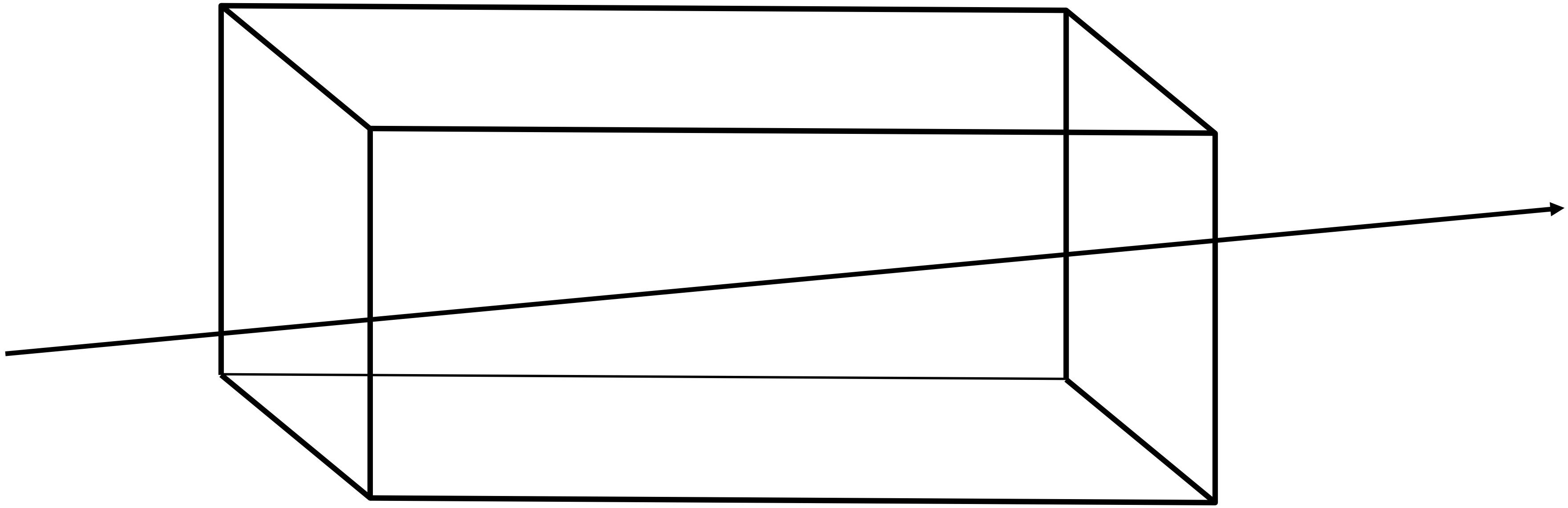
McStas magnetic fields

- `Pol_constBfield.comp`
- Single constant Magnetic field in a “box”.
- - user may specify a wavelength to flip.
  - blocking walls



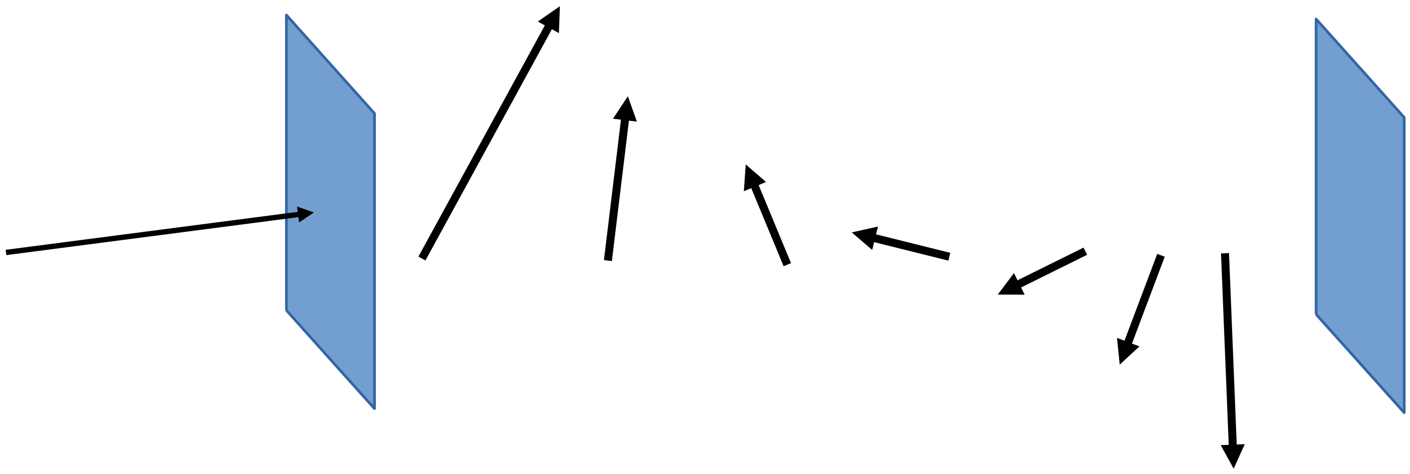
McStas magnetic fields

- `Pol_FieldBox.comp`
- **Single Magnetic field in a "box"**
- **- optional user supplied field c-function**



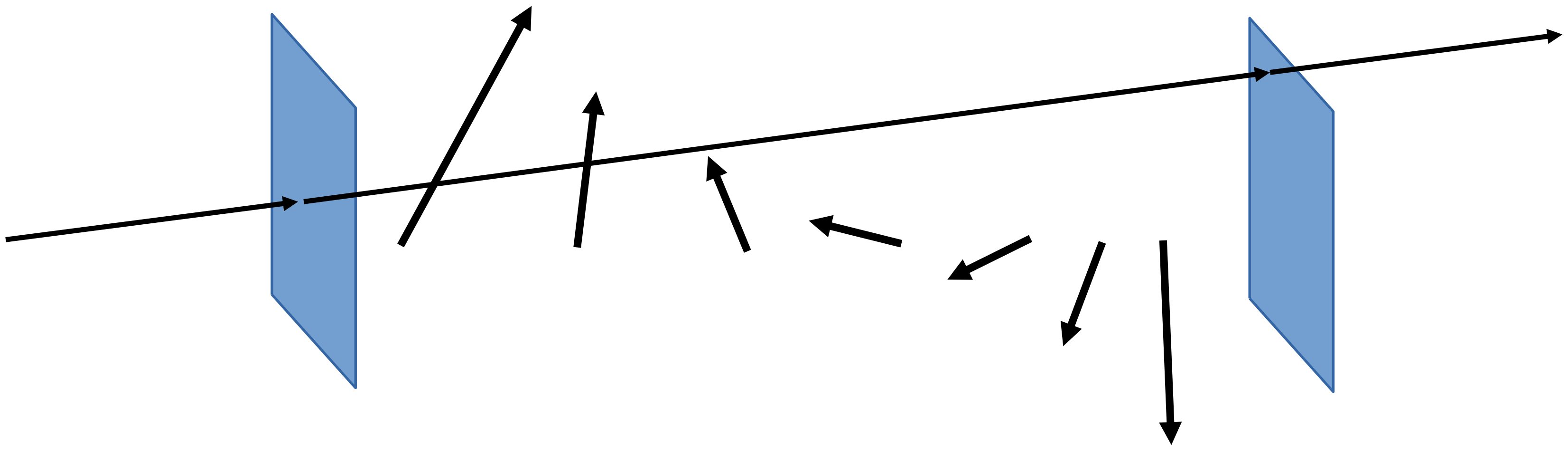
McStas magnetic fields

- `Pol_simpleBfield.comp`
- `Pol_simpleBfield_stop.comp`
    - - Entry/Exit contruction allows for nested magnetic field descriptions.
      - Any magnetic fields through user supplied c-function
      - Tabled magnetic fields

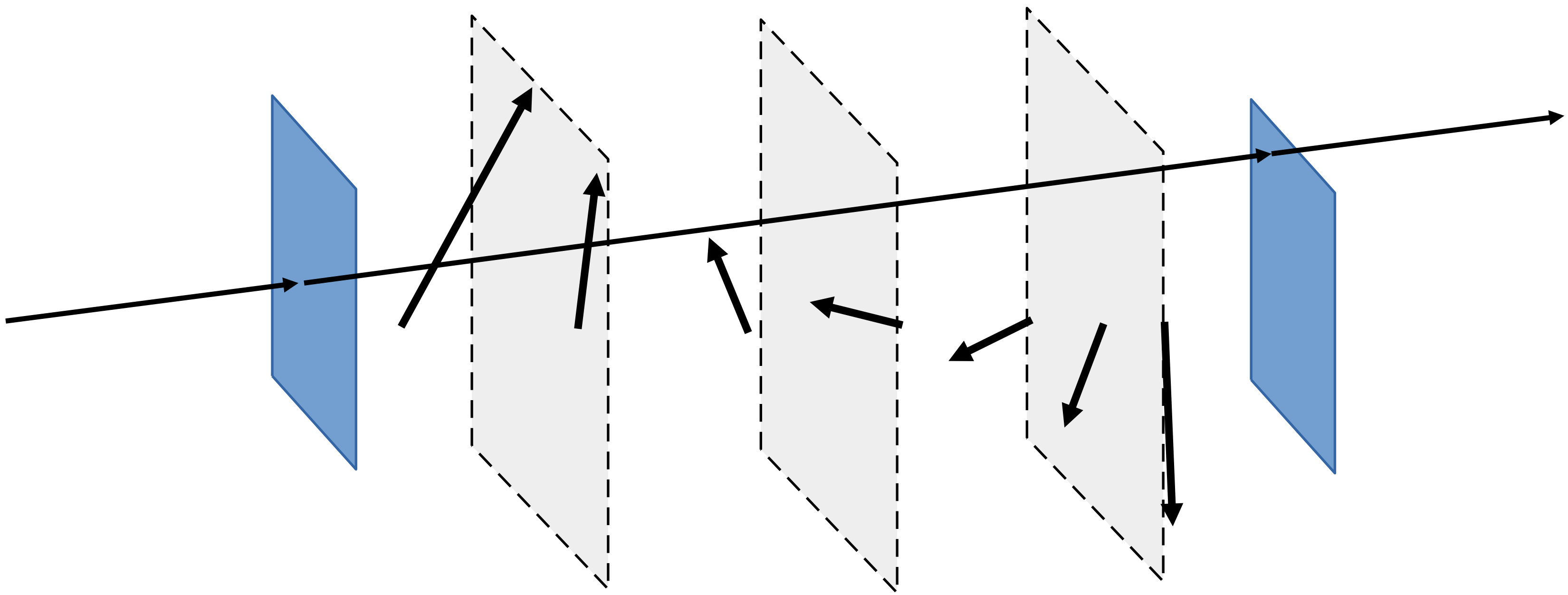


McStas Polarization Capabilities IV

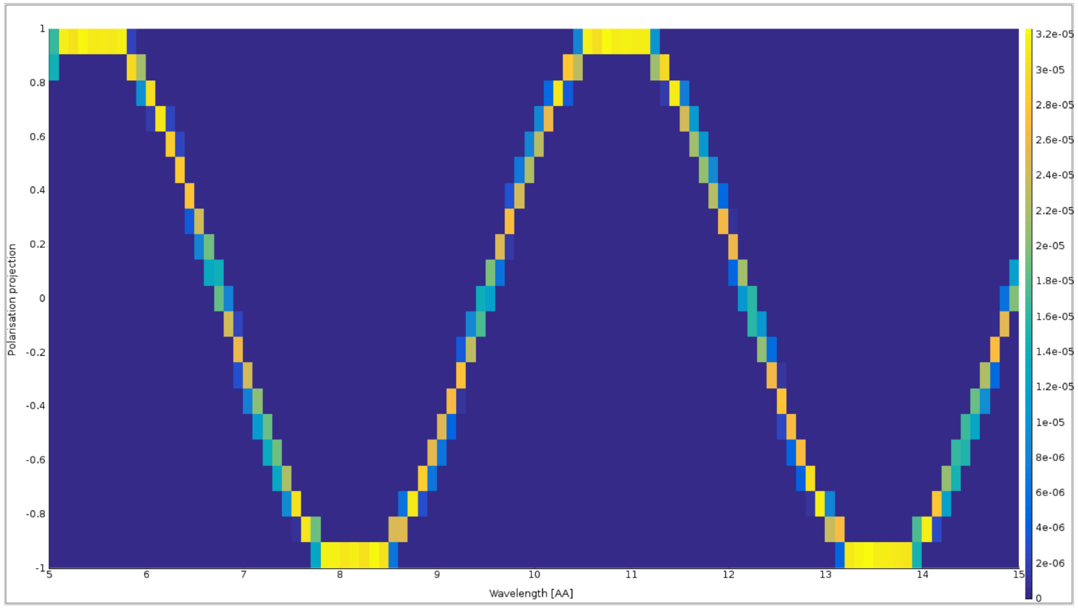
- `Pol_simpleBfield.comp`
- `Pol_simpleBfield_stop.comp`
  - - Entry/Exit contruction allows for nested magnetic field descriptions.
    - Any magnetic fields through user supplied c-function
    - Tabled magnetic fields

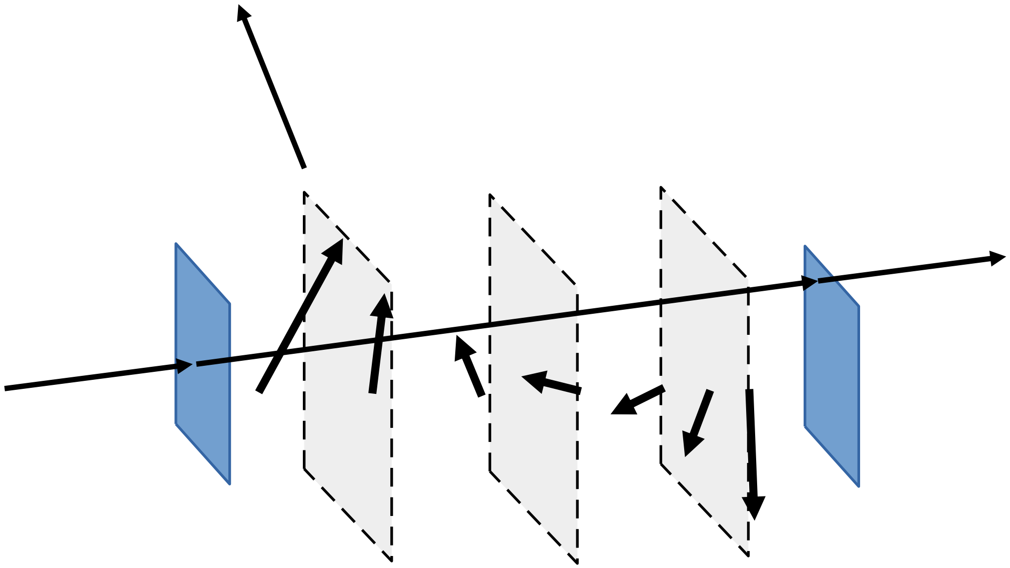


Pol_monitors along the way...

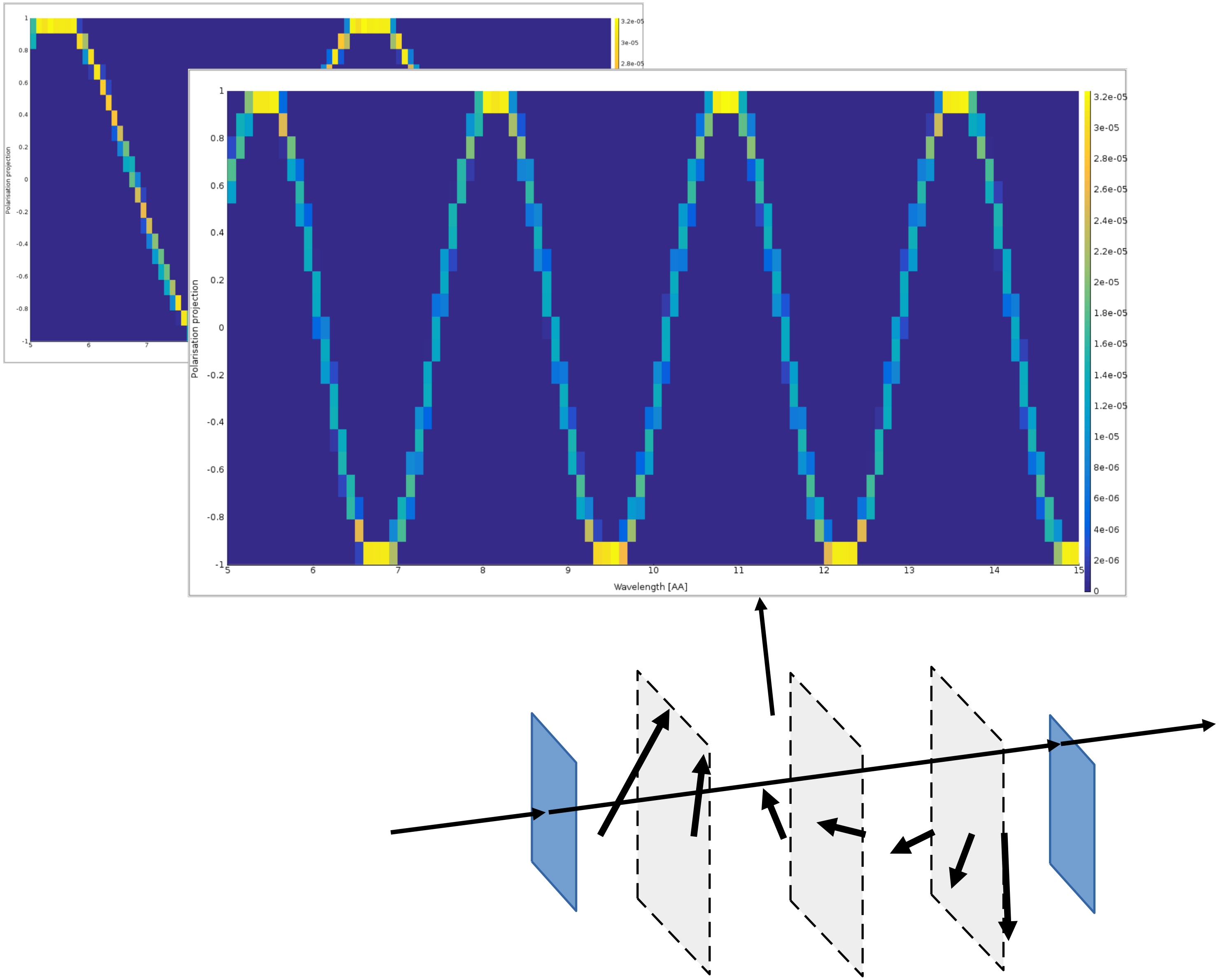


Pol_monitors along the way...

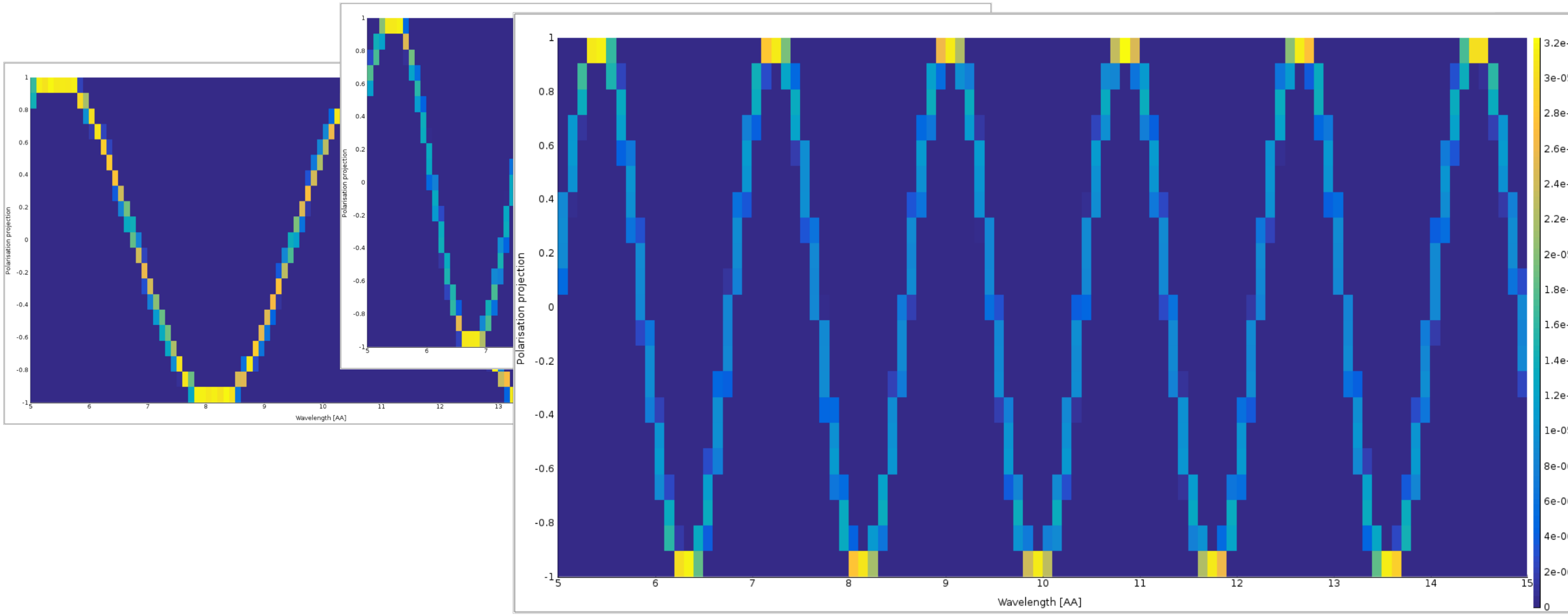


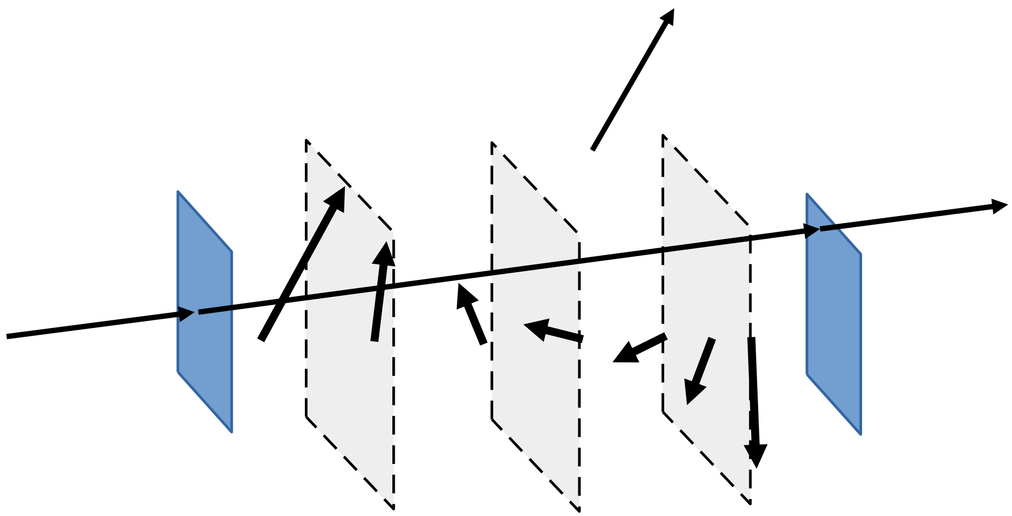


Pol_monitors along the way...

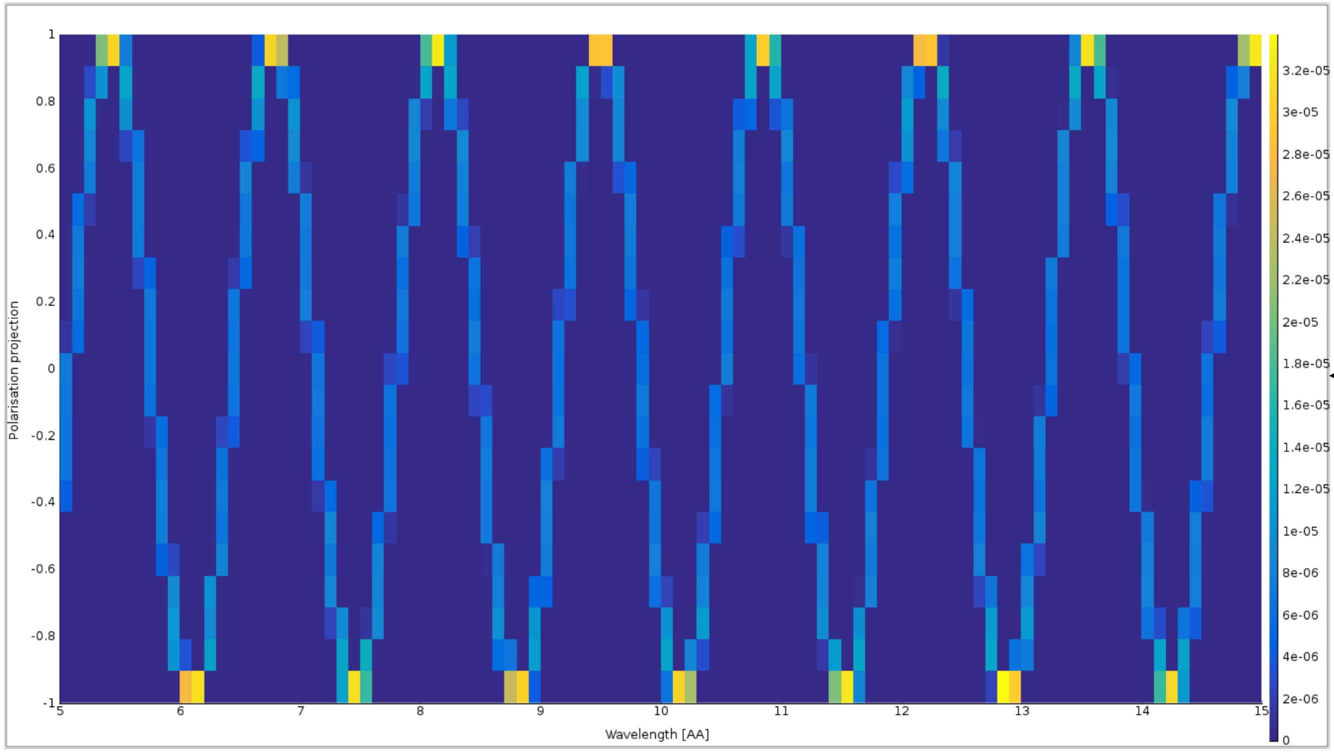


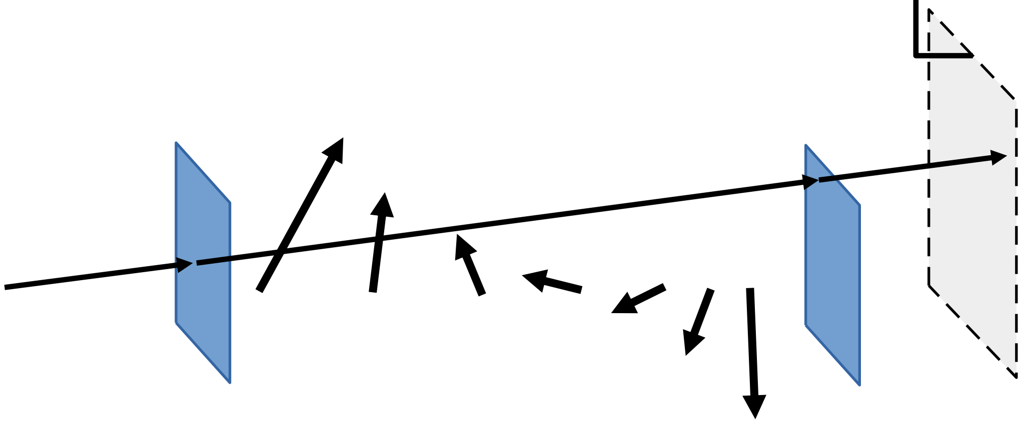
Pol_monitors along the way...



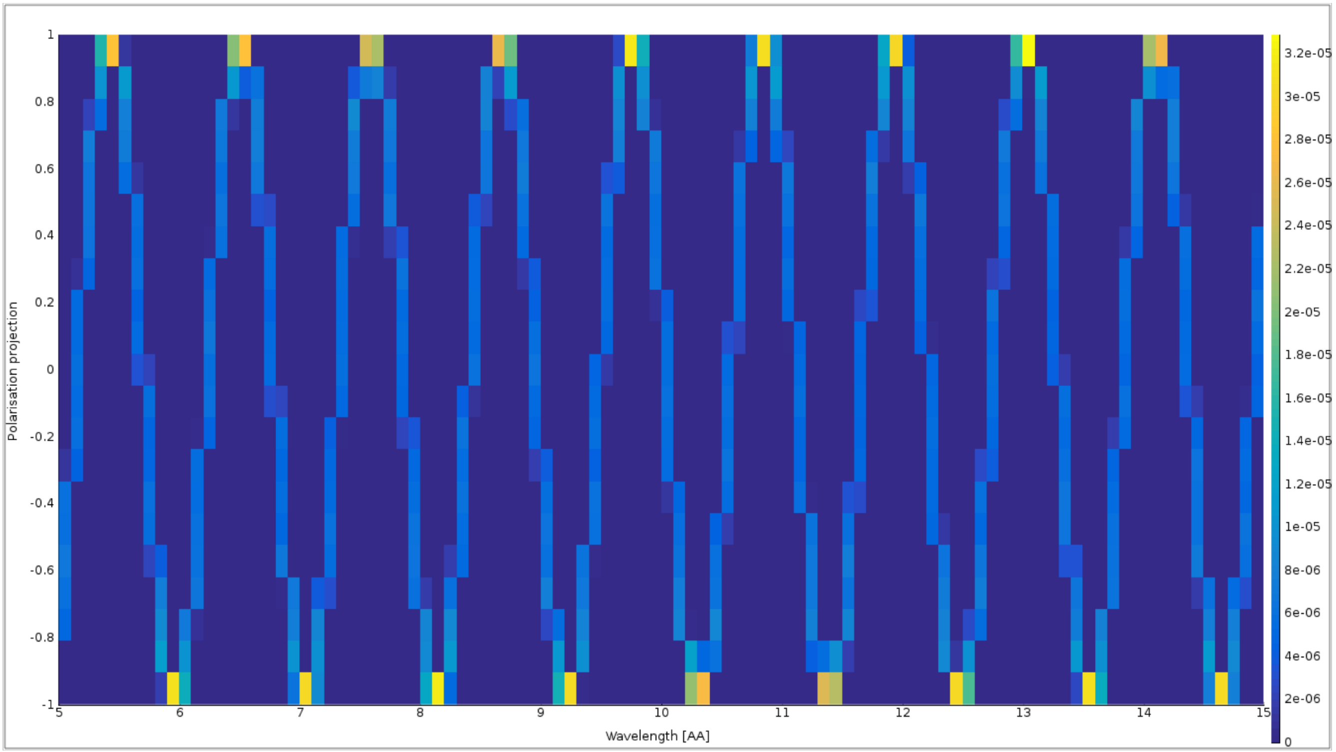


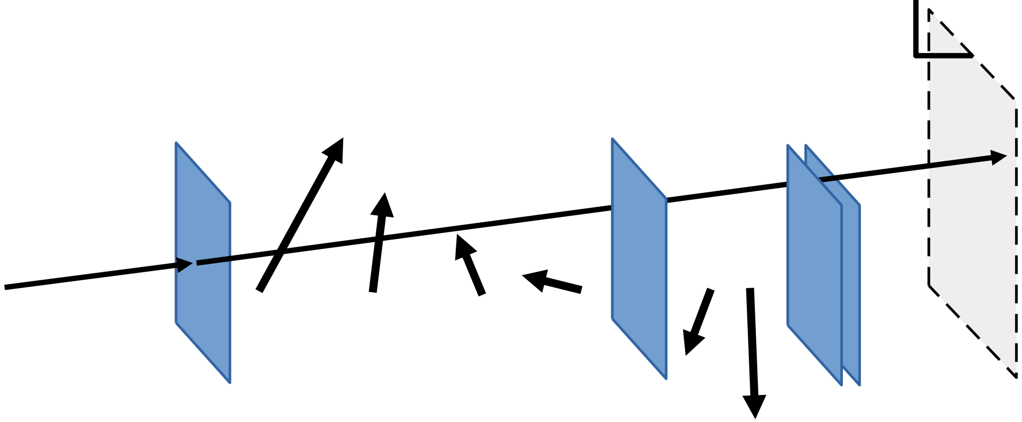
Nested fields





Nested fields





How does one go about it?

Example walk-through: SE-template

How does one go about it?

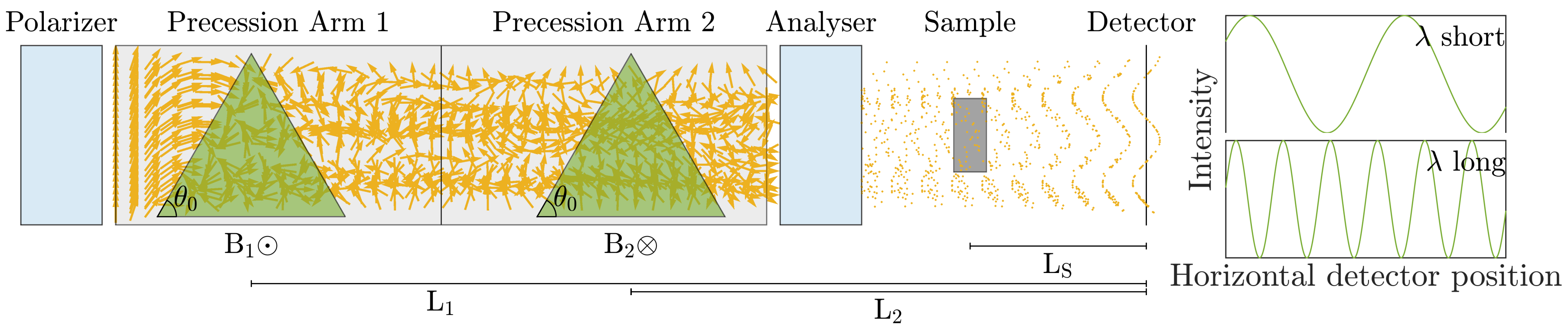
Example walk-through: SE-template

Getting help

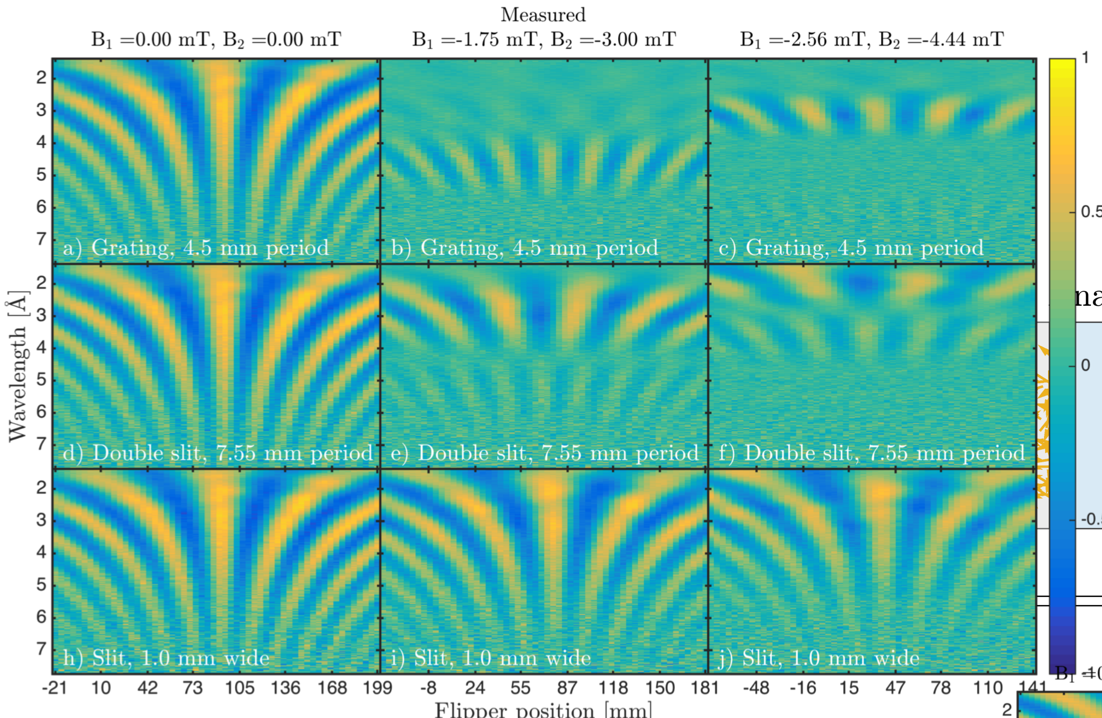
- Check example header.
- Use `mcdoc`
- Read/check the manual
- User mailing list: `mcstas-users@mcstas.org`
- Give us a call/write us an email!

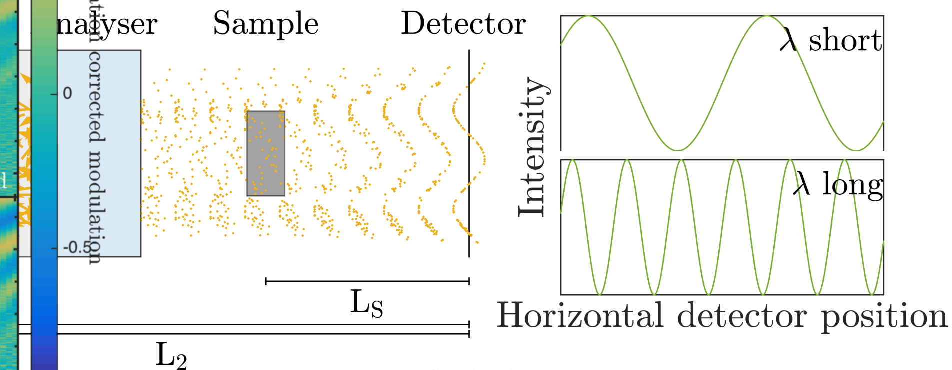
McStas example SEMSANS

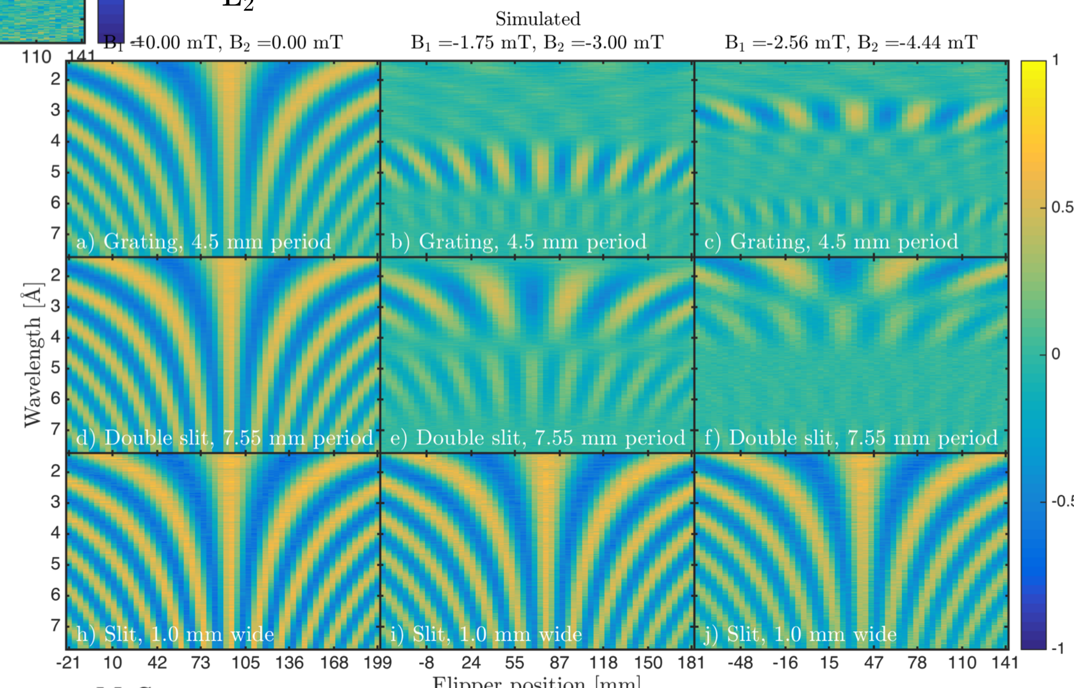
Courtesy: M. Sales et.al.



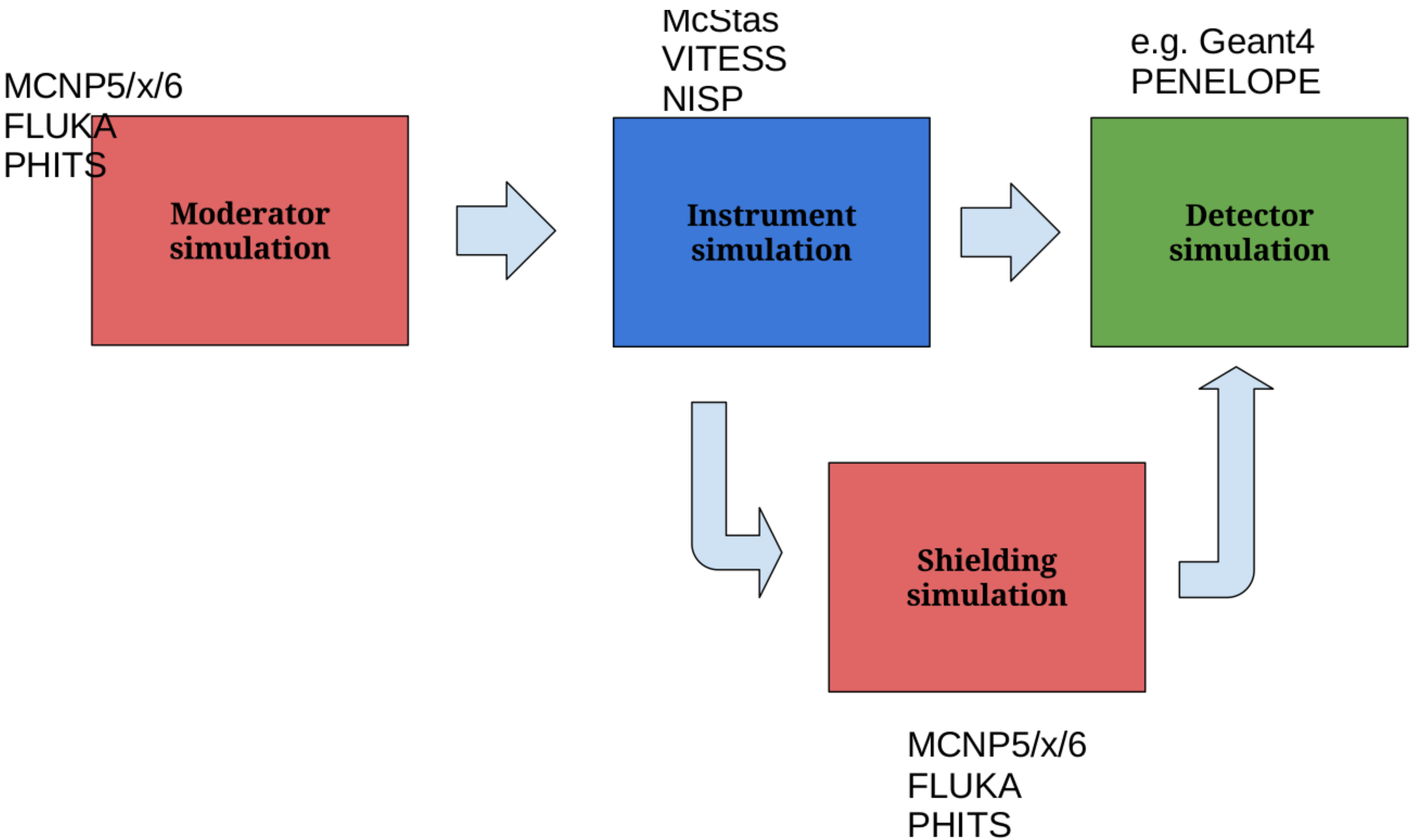
McStas example SEMSANS



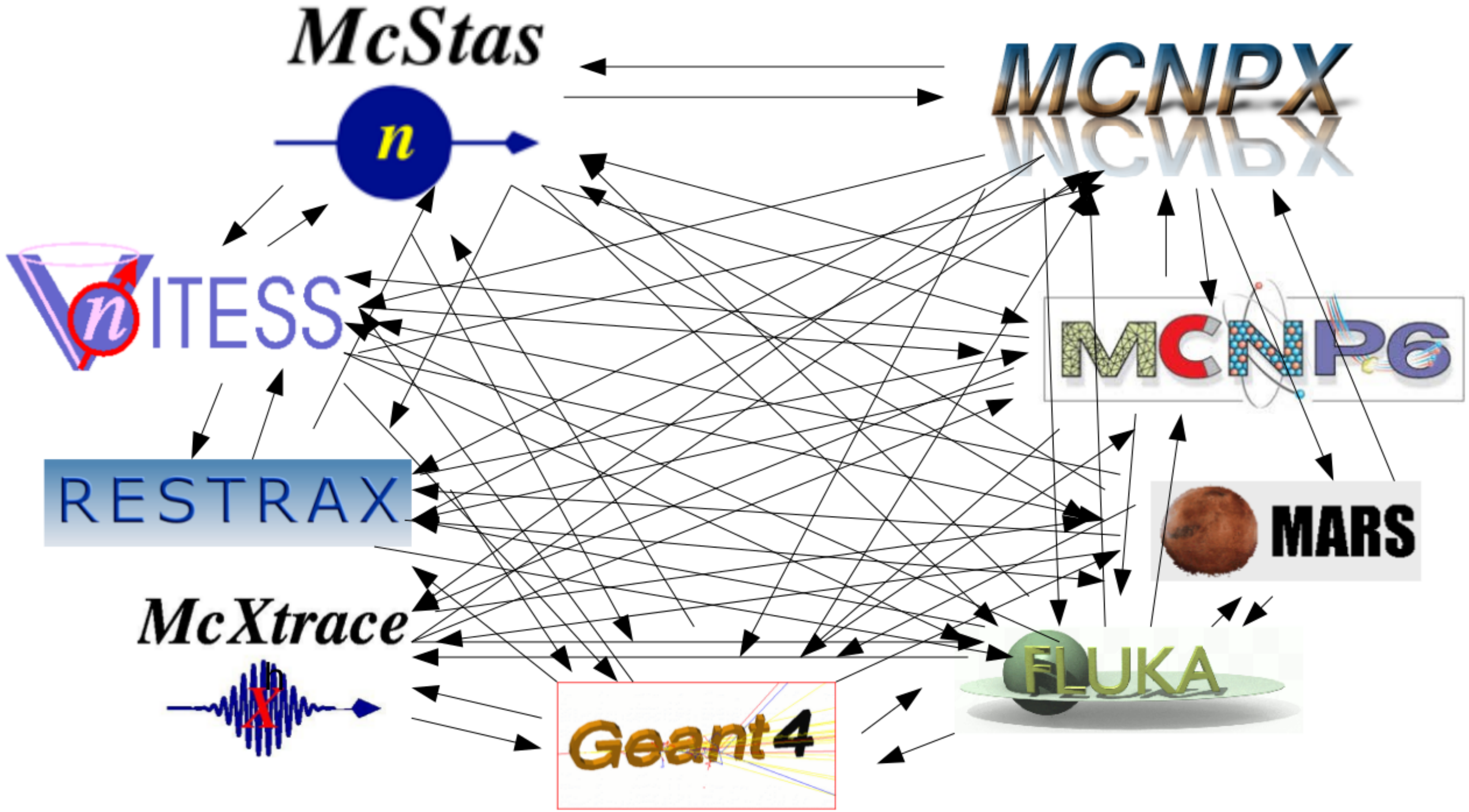




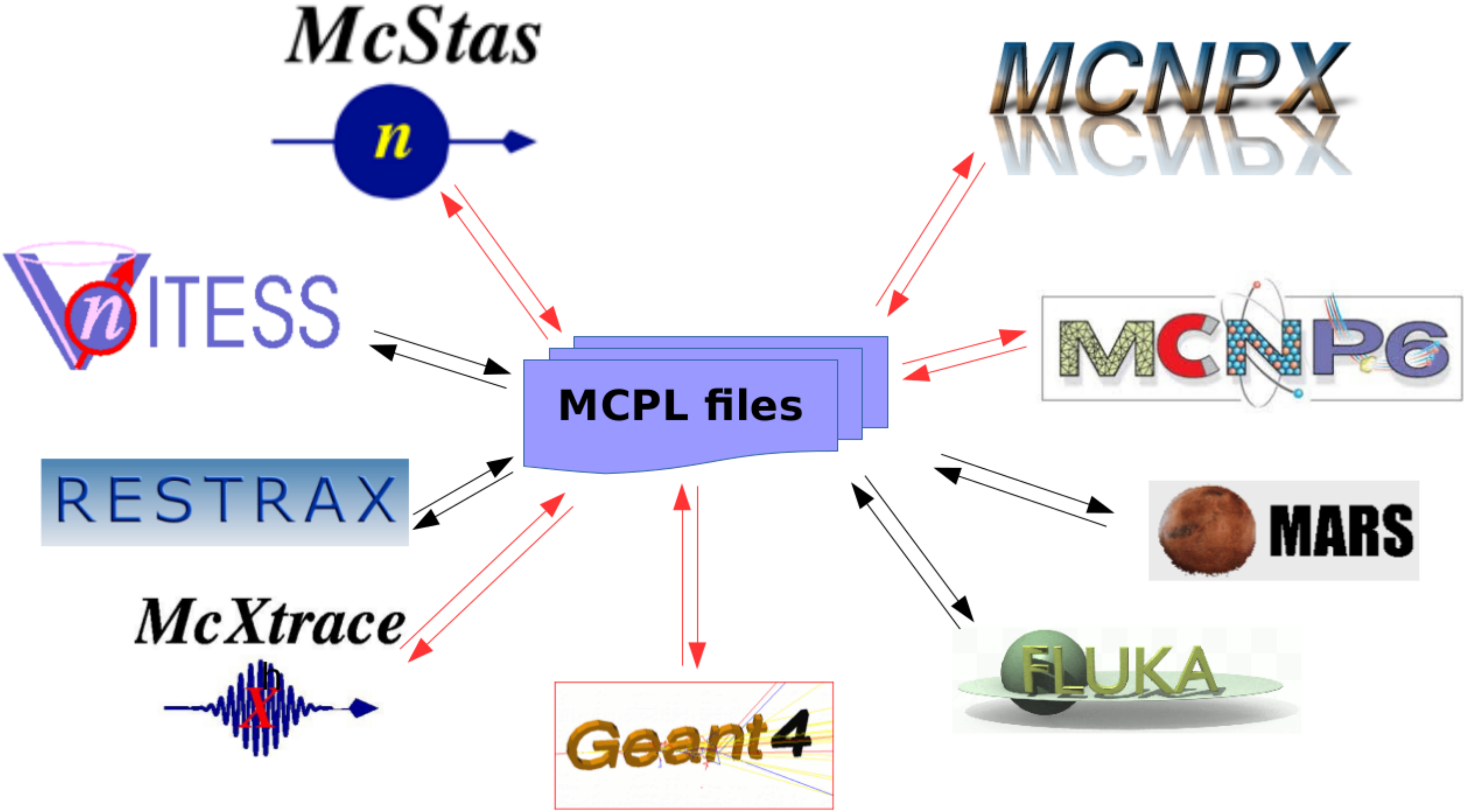
A goal: real sims with background



The problem


McStas
MCNPX
VITESS
MCNP6
RESTRAX
MARS
McXtrace
FLUKA
Geant4

The solution



McStas components on the way

Things on the way

Magnetic fields:

- Pol_FieldBox.comp
  - Tabled fields
- Pol_constBfield.comp
- Pol_simpleBfield.comp
  - 3D entry/exit windows
- Pol_simpleBfield_stop.comp
- Pol_triafield.comp

Optics:

- Monochromator_pol.comp
- Pol_bender.comp
- Pol_guide_vmirror.comp
- Pol_mirror.comp
- Pol_pi_2_rotator.comp
- Transmission_polarisatorABSnT.comp
- Pol_bender_tapering.comp
- Pol_McRadia.comp
  - Dynamic coupling to RADIA

Monitors:

- Pol_monitor.comp
- MeanPolLambda_monitor.comp
- PolLambda_monitor.comp
- Pol_PSD_monitor.comp

Idealized components:

- PolAnalyser_ideal.comp
- Set_pol.comp
- Pol_SF_ideal.comp

Contrib:

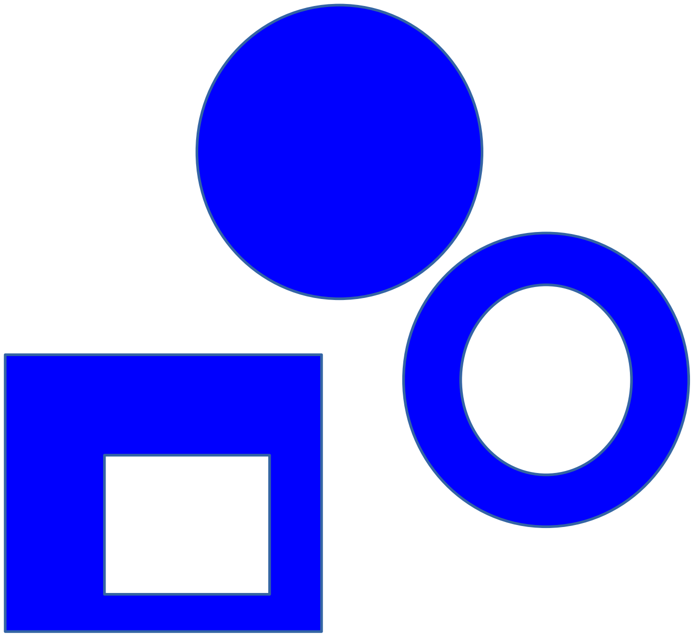
- Foil_flipper_magnet.comp

Sample component

- Magnetic_single_crystal.comp

McStas components on the way

Generalized Simple B-Fields: constant, functional, tabled, … but in more general shapes



RF-flipper

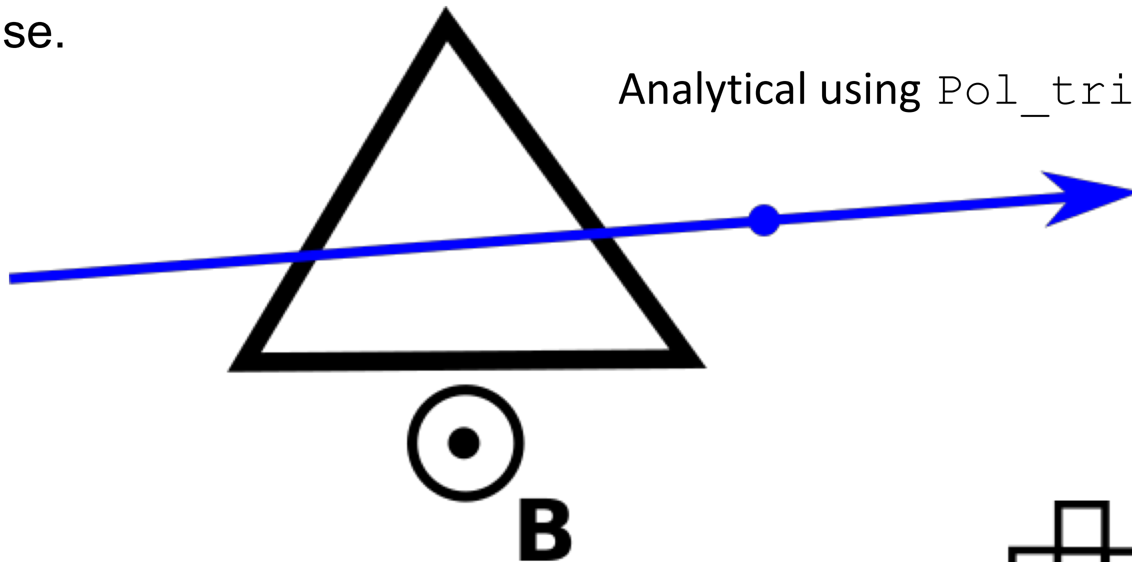
He3-objects

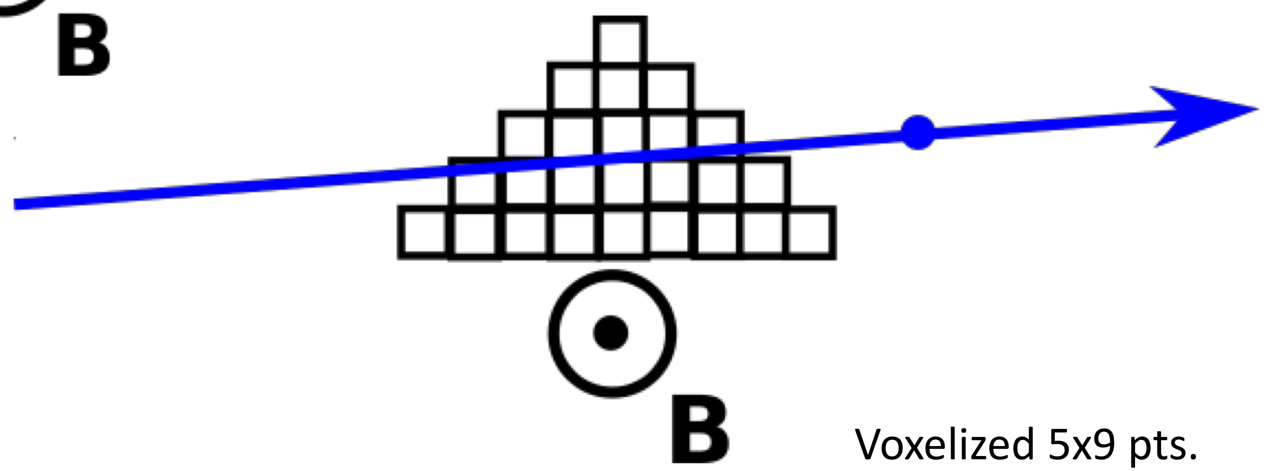
McStas components on the way

McRadia compared with analytical field description

Requires a mathematica license.

Analytical using `Pol_triafield.comp`

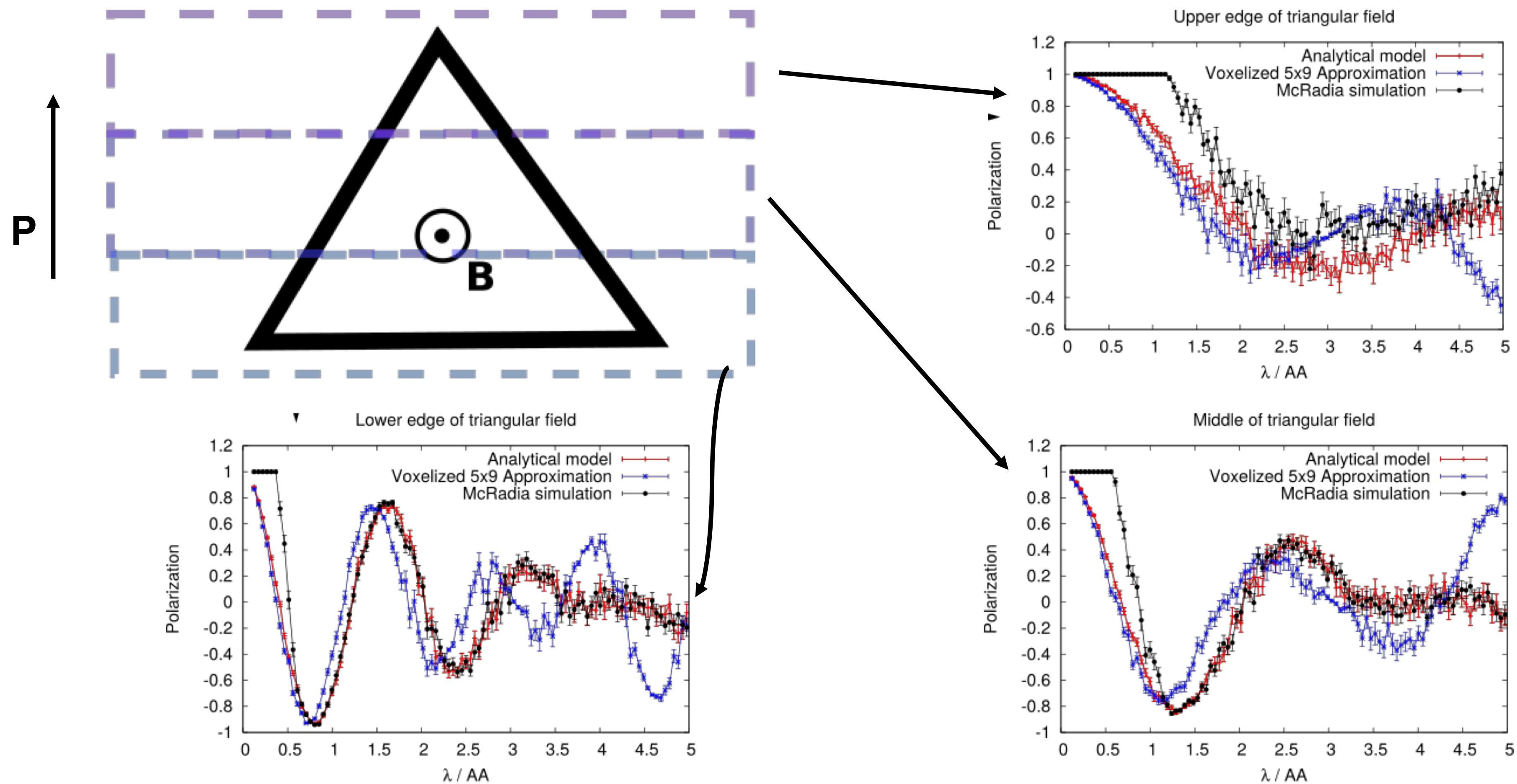




Voxelized 5x9 pts.

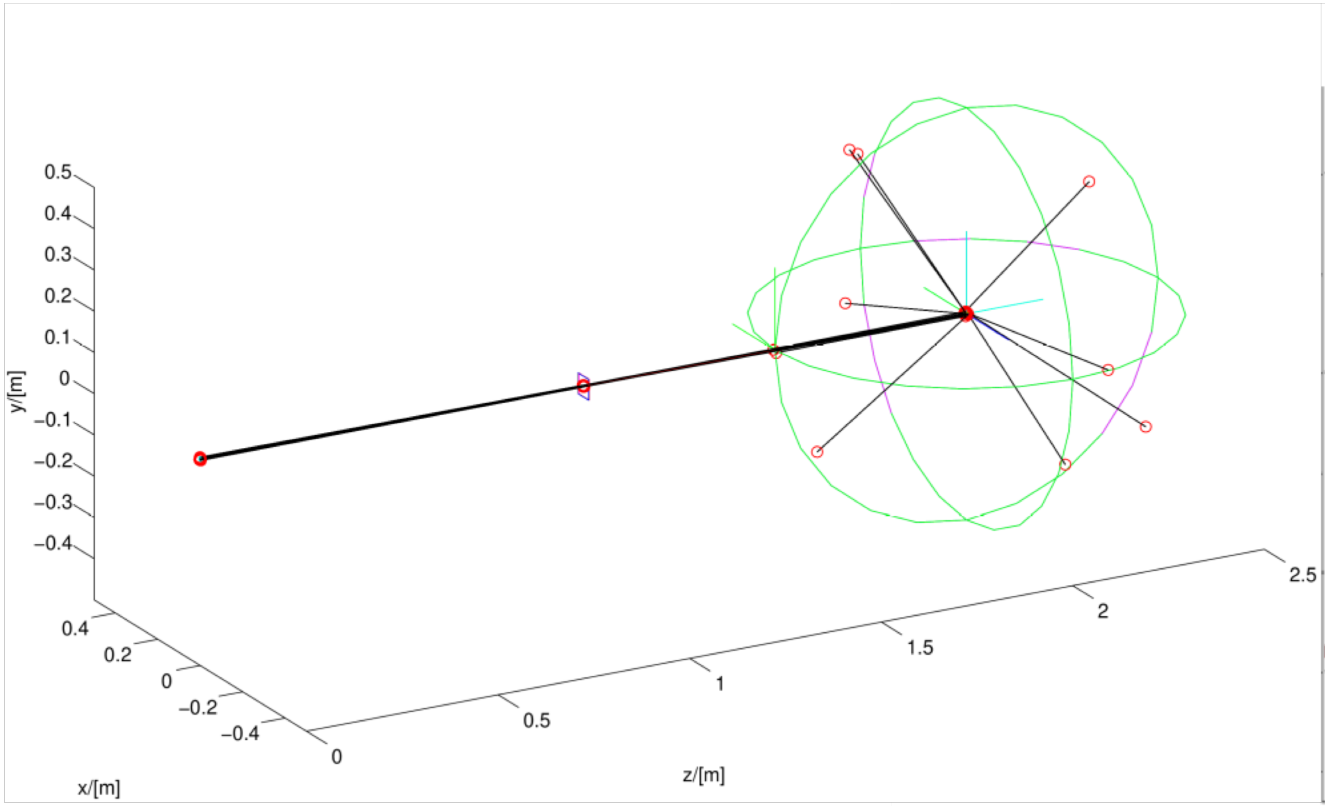
McStas components on the way

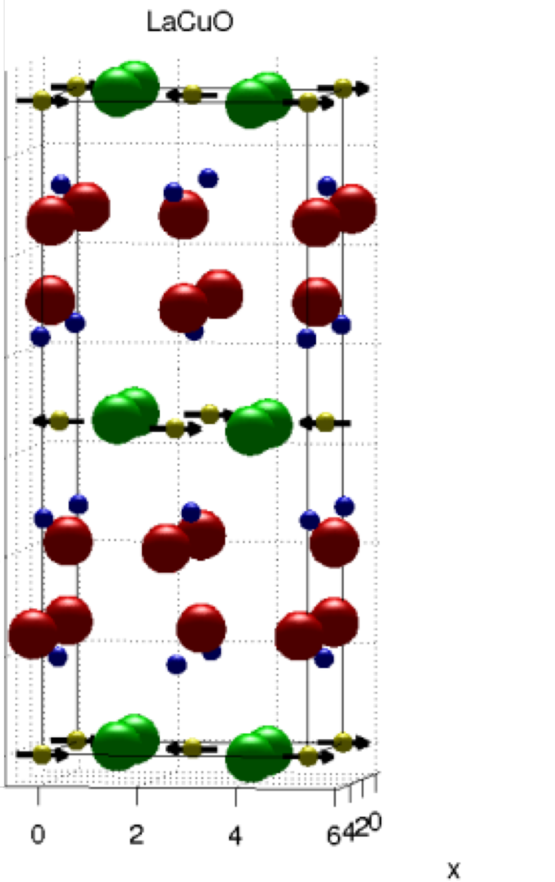
McRadia compared with other field descriptions



McStas components on the way

Magnetic single crystal

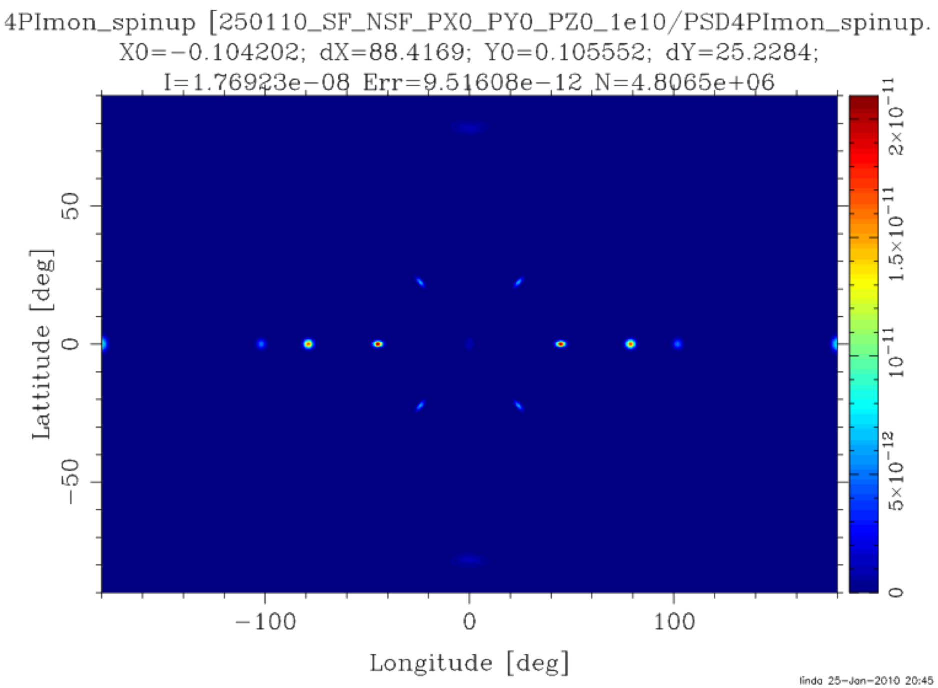


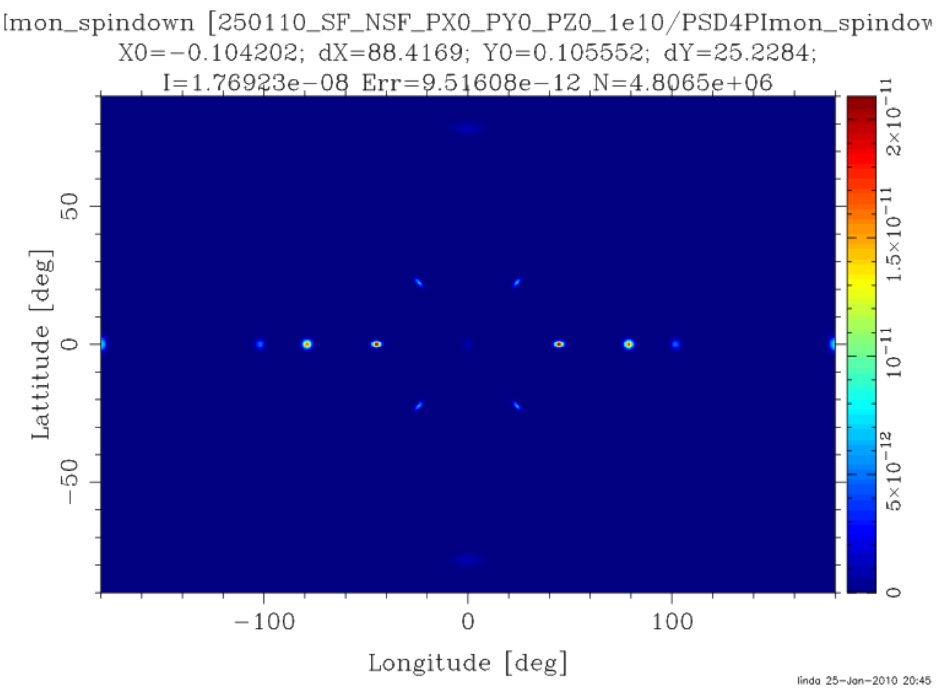


index	iontype	x	y	z	b_{coh}[fm]	g_S	S_x	S_y	S_z	g_L	L_x	L_y	L_z
1	Cu2+	0.5	0.5	0	7.718	2	0	-0.5	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

McStas components on the way

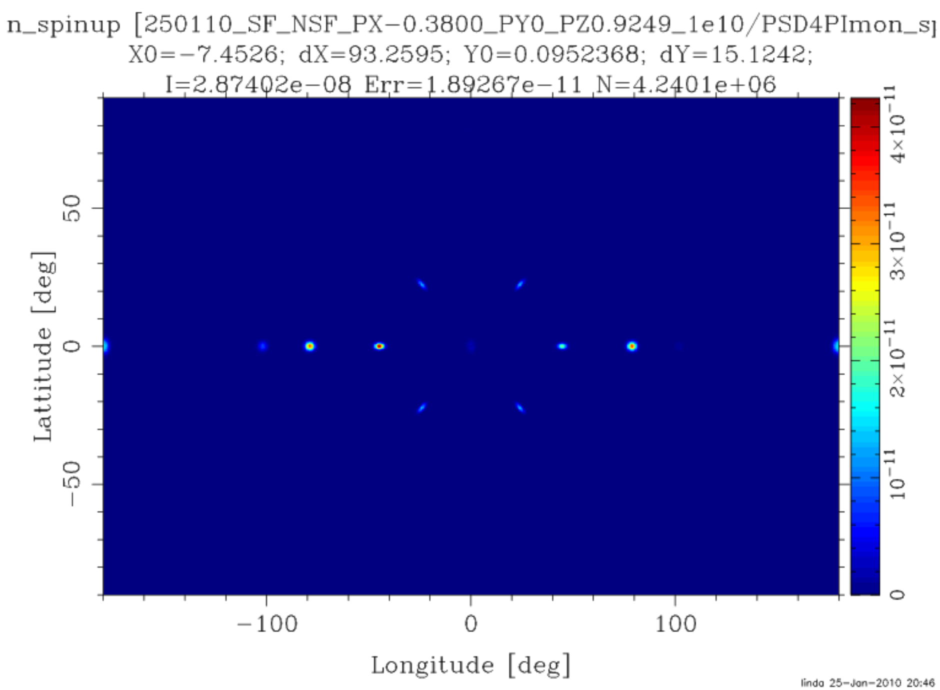
Magnetic single crystal – Unpolarized beam

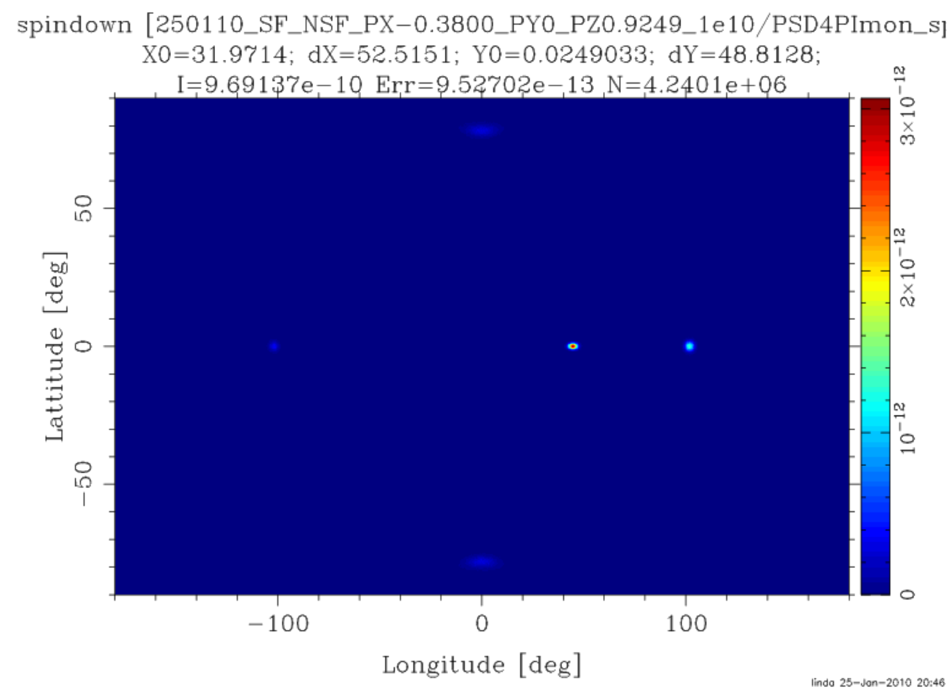




McStas components on the way

Magnetic single crystal – Polarized beam





McStas components on the way

Magnetic single crystal

The magnetic scattering cross-section for a sample with localised spin+orbital angular moment $g\mathbf{J} = (g_S + g_L)\mathbf{J} = 2\mathbf{S} + \mathbf{L}$ is:

$$\frac{\mathrm{d}^2\sigma}{\mathrm{d}\Omega_\mathrm{f}\mathrm{dE}_\mathrm{f}} = \frac{k_f}{k_i}\sum_{i,f} P(\lambda_i)\left|\langle\lambda_f \mid \sum_j e^{i\mathbf{Q}\cdot\mathbf{d}_j}\, U_j^{\sigma_i\sigma_f} \mid \lambda_i\rangle\right|^2 \delta(\hbar\omega + E_i - E_f)$$

where $\mid \lambda_i\rangle$ and $\langle\lambda_f \mid$ are the initial and final states of the sample with energies E_i and E_f respectively, $P(\lambda_i)$ is the distribution of initial states and

$$U_j^{\sigma_i\sigma_f} = \langle\sigma_f \mid b_j - m_j\mathbf{J}_{\perp j}\cdot\boldsymbol{\sigma} \mid \sigma_i\rangle$$

where $\mid \sigma_i\rangle$ and $\langle\sigma_f \mid$ are the initial and final spin states of the neutron, and $\boldsymbol{\sigma}$ are the Pauli spin matrices working on the neutron state.

From: G. Shirane et.al. ,"Neutron Scattering with Triple-Axis Spectrometer", *Cambridge Univ. Press*, 2002

Magnetic single crystal

If $\mathbf{P} = P(\xi, \eta, \zeta) = P\hat{\zeta}$.Thus, the matrix elements of $U^{\sigma_i \sigma_f}$ can now be written

$$\begin{aligned} U^{++} &= b - mJ_{\perp\zeta} \\ U^{--} &= b + mJ_{\perp\zeta} \\ U^{+-} &= -m\left(J_{\perp\xi} + iJ_{\perp\eta}\right) \\ U^{+-} &= -m\left(J_{\perp\xi} - iJ_{\perp\eta}\right) \end{aligned}$$

where $m = \frac{r_0 \gamma}{2} g f(\mathbf{Q})$ with r_0 the classical electron radius, $\gamma = 1.913$, g the Landé splitting factor and $f(\mathbf{Q})$ the magnetic form factor of a particular ion in the sample.

Acknowledgements

DTU

Peter Willendrup
Jacob Garde

ILL

Emmanuel Farhi

Copenhagen University

Kim Lefmann
Mads Bertelsen

PSI

Emmanouela Rantsiou
Uwe Filges

Kristian Nielsen
Kurt Clausen
Peter Christianse
Klaus Liutenant

And all other contributors!

Outlook/Discussion items

- How can we interact better?
  - Better support for the community?
  - Code-sharing?