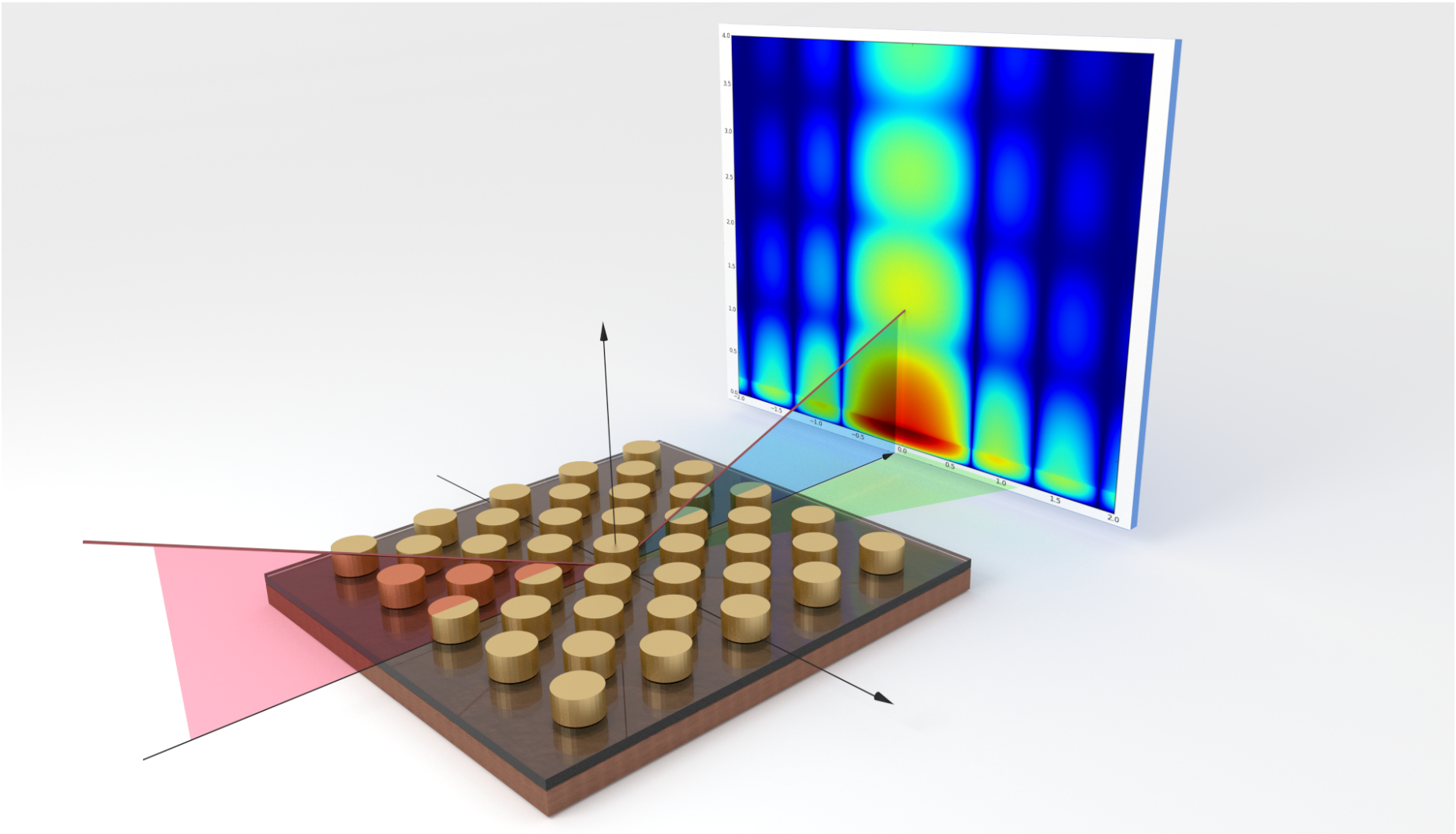
BornAgain Essentials

Jan Burle, Jonathan M. Fisher, Marina Ganeva, Gennady Pospelov, Walter Van Herck and Joachim Wuttke

Overview

- Experimental setup: beam, sample, detector (binning)
- Feature set
- Software API architecture
- Coordinate systems
- Development infrastructure

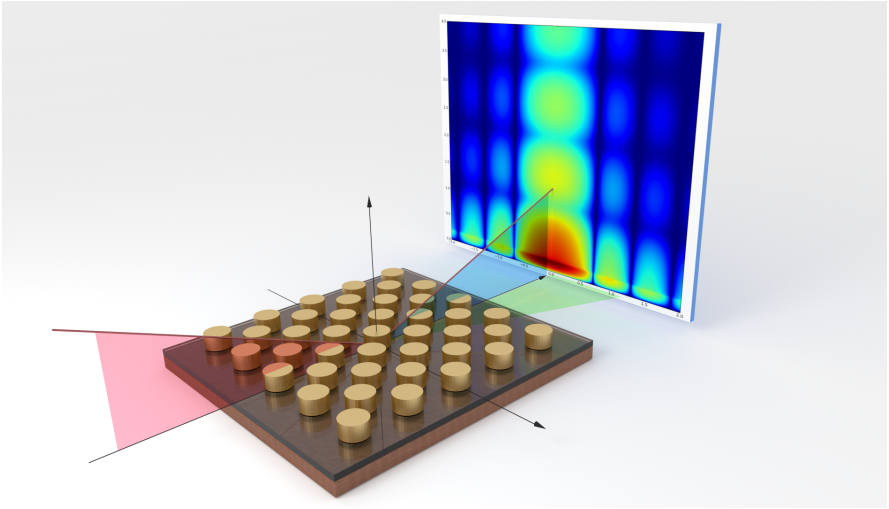
Experimental setup



GISAS specifics

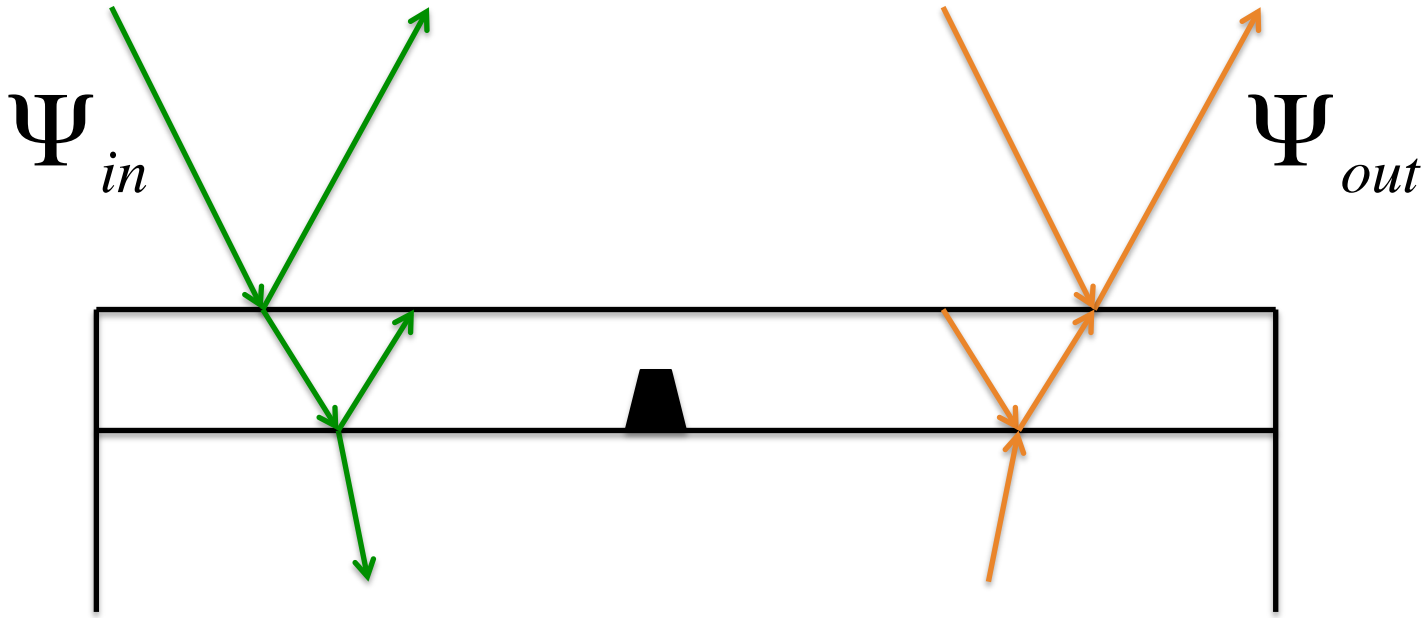
- In-plane and out-of-plane information
- Increased scattering volume
- Reflected and transmitted waves interfere
- Software can help to arrive at quantitative results (simulate & fit)

$$V_{scat} \sim 2\frac{A_{beam} \cdot t}{\sin \alpha_i}$$



Distorted Wave Born Approximation

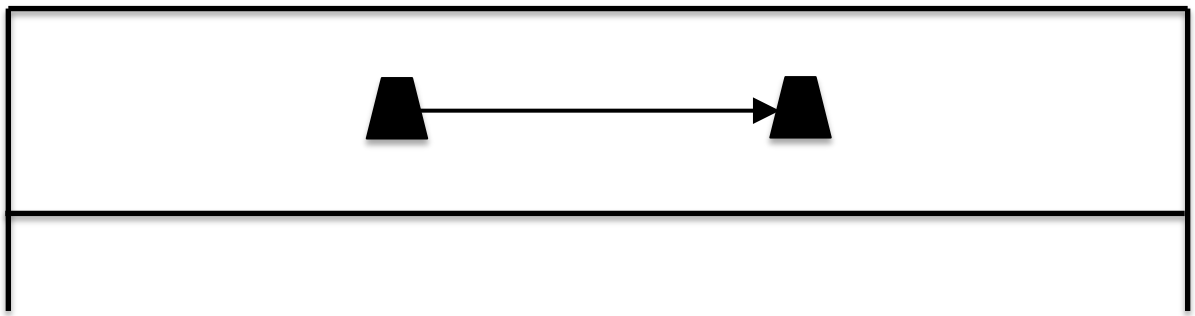
- Unperturbed system: smooth multilayer
  Solutions: reflected and transmitted plane waves
- 1st order scattering on embedded nanoparticles: 4 terms
- Similar with roughness



Shape and position

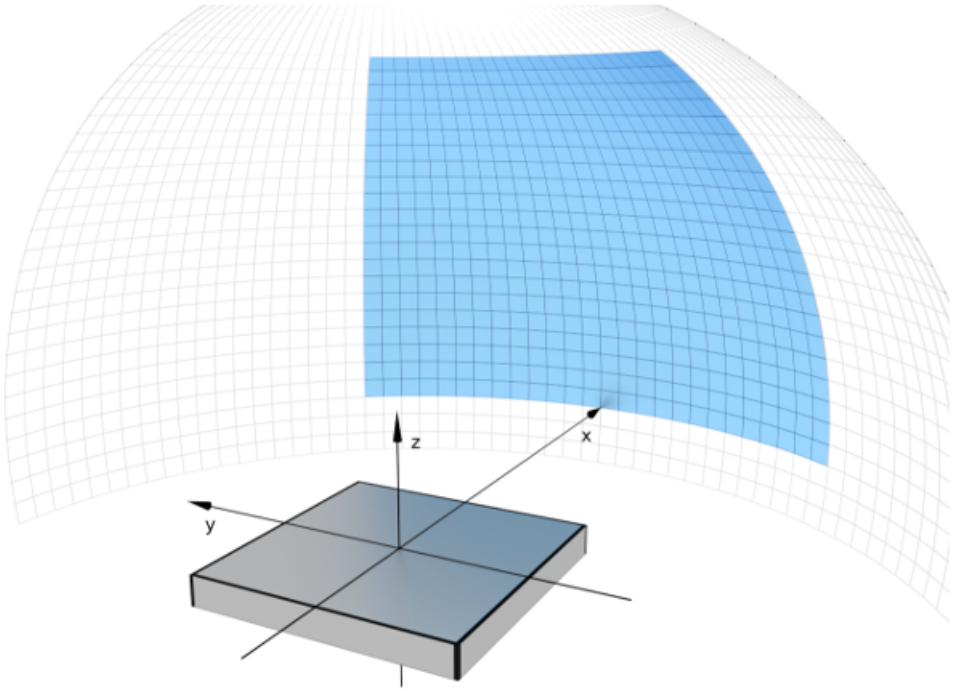
- Shape: form factor
- Positions: interference function

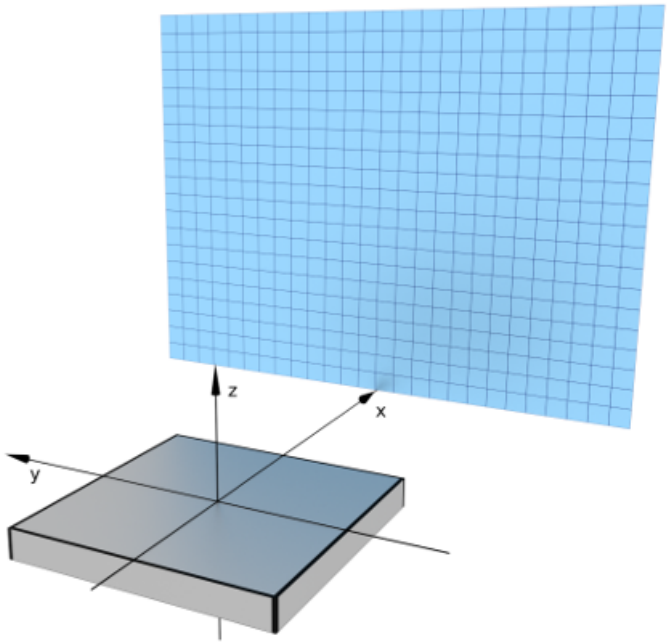
$$N\frac{d\sigma}{d\Omega} = \left|\sum_i \exp(iq\cdot R_i)F_{DWBA}(q;T_i)\right|^2$$



Beam, detector

- Idealized beam: monochromatic, no angular divergence
  Real beam: model with distributions / resolution function
- Spherical or rectangular detector



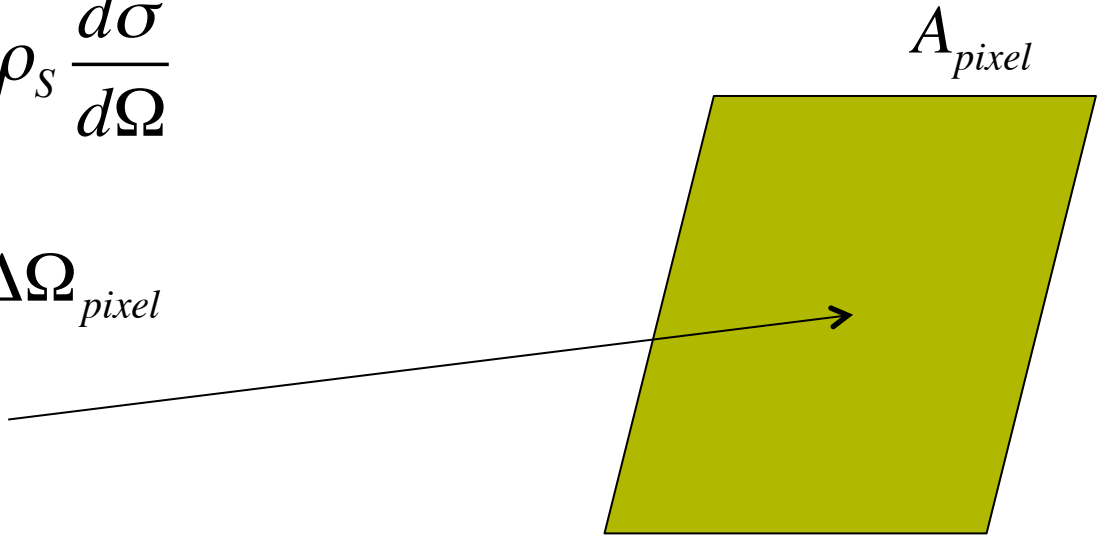


Intensity in detector pixel

- If pixel is small enough, value at center of pixel times its size
- Integration of differential cross-section (Monte Carlo)

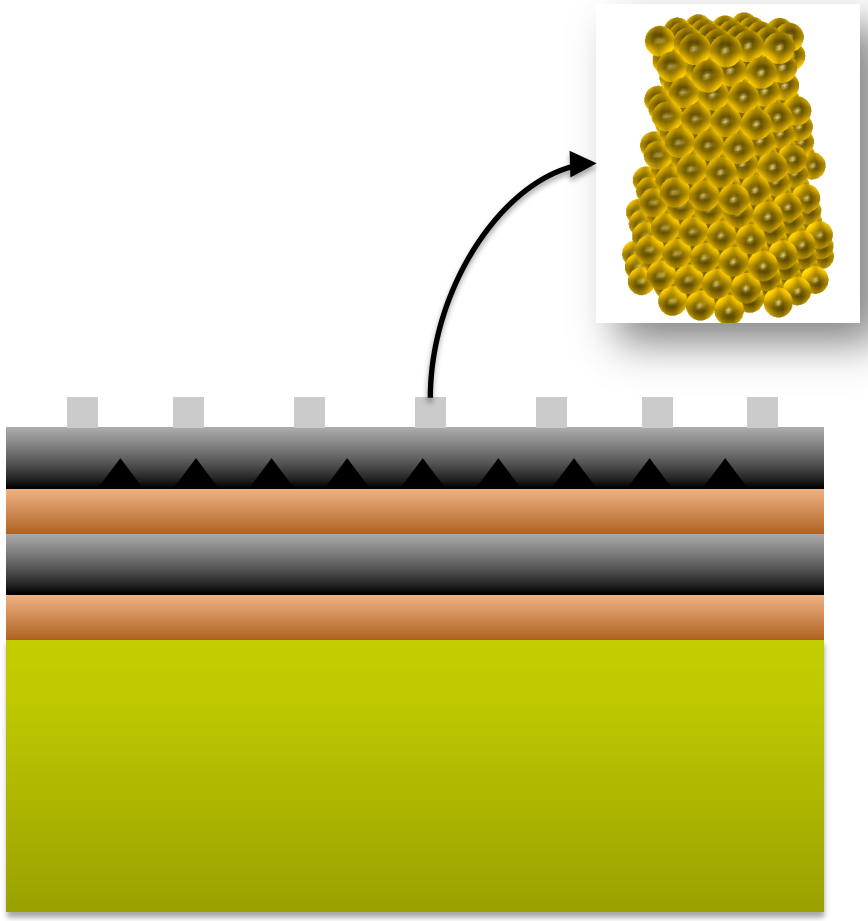
$$I_{pixel} = \int_{A_{pixel}} d\Omega \frac{I_0}{\sin\alpha_i} \rho_S \frac{d\sigma}{d\Omega}$$

$$\cong \frac{I_0}{\sin\alpha_i} \rho_S \frac{d\sigma}{d\Omega} \Delta\Omega_{pixel}$$



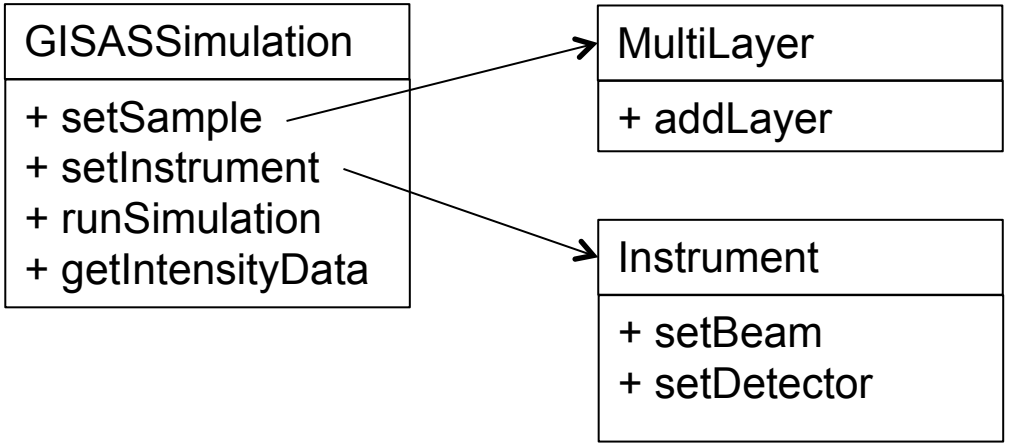
Core feature set

- Multilayer
- Interface roughness
- Multiple nanoparticles (shapes, densities)
- Interference functions
- Nanoparticles assemblies
- Mesocrystals
- Polarized neutron scattering



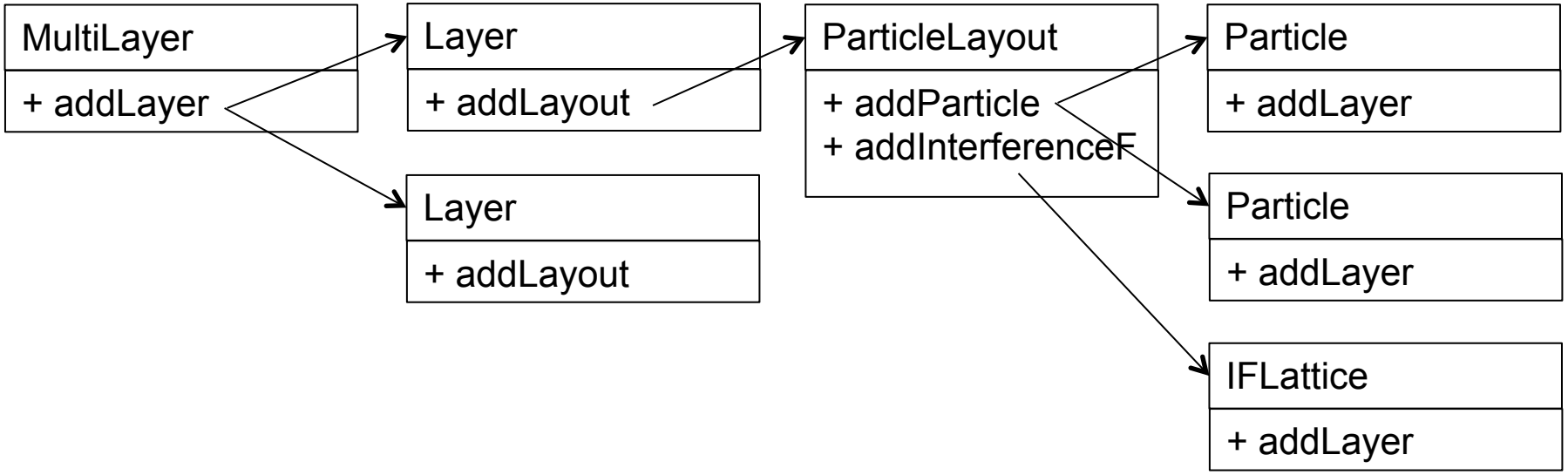
Structure of software API: simulation

- GISASSimulation contains sample and instrument
- Additionally, it may contain distributions, builders, ...
- runSimulation calculates scattering intensity
- getIntensityData retrieves the intensity map



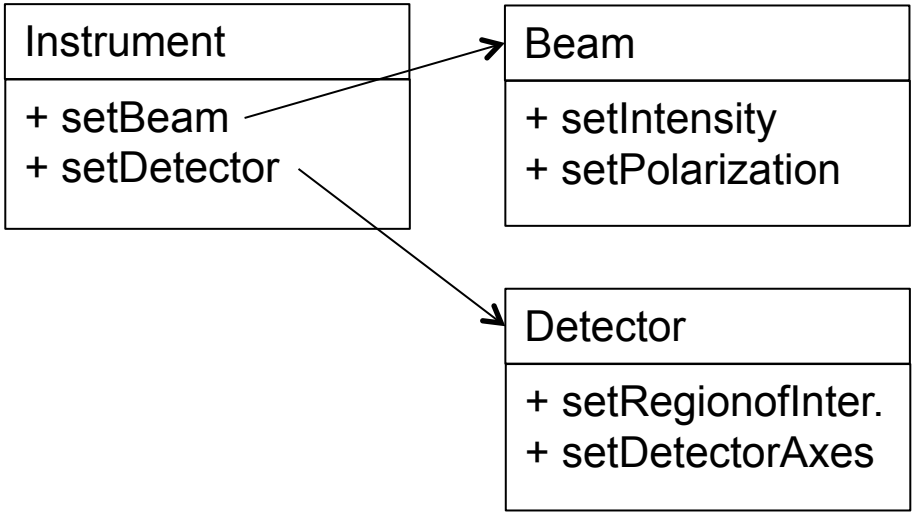
Structure of software API: sample tree

- Root of the tree is a multilayer
- Multiple layers possible
- Multiple layouts possible
- Multiple particles possible ...



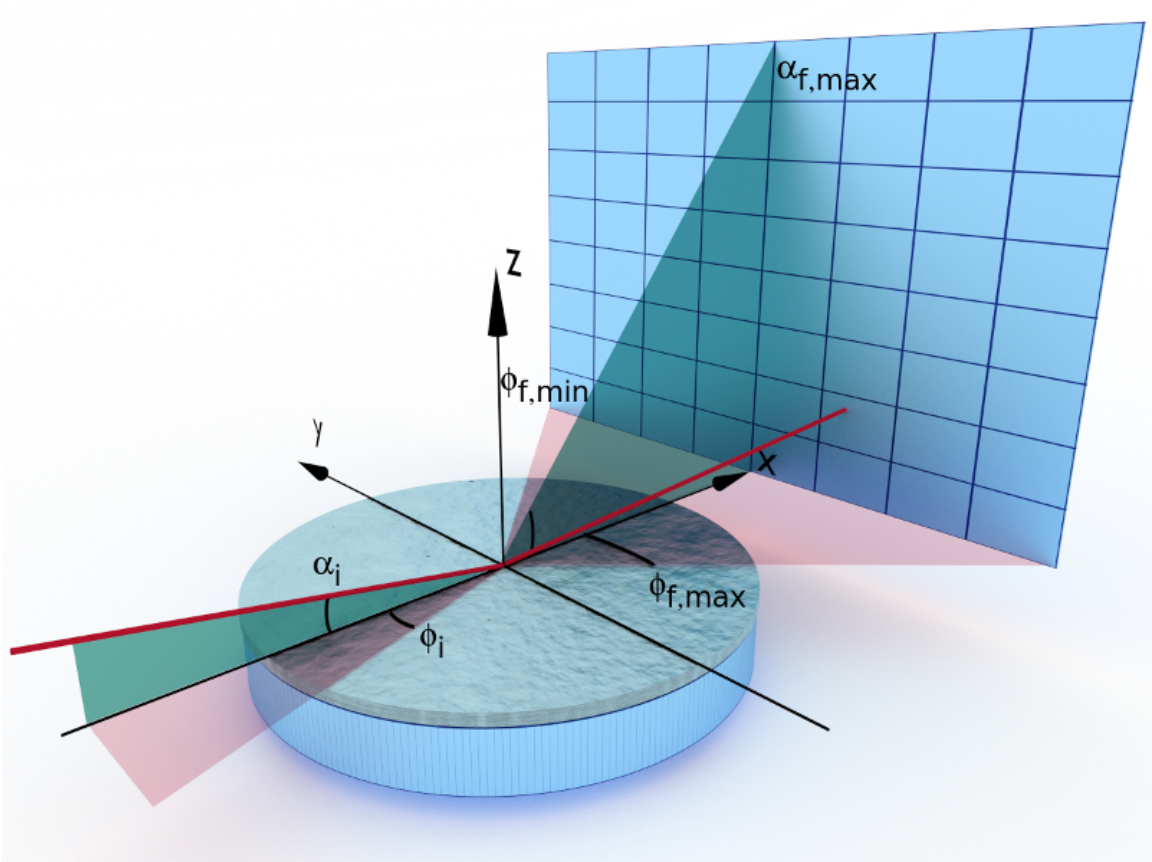
Structure of software API: instrument

- Instrument contains beam and detector
- Beam: wavelength, direction, intensity, polarization
- Detector: type, size, position, masks, ROI, ...



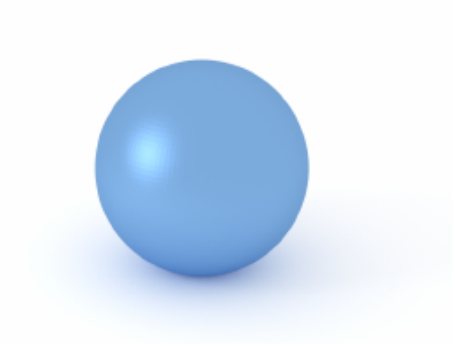
Coordinate systems: global

- Global coordinate system
- z-axis pointing upwards

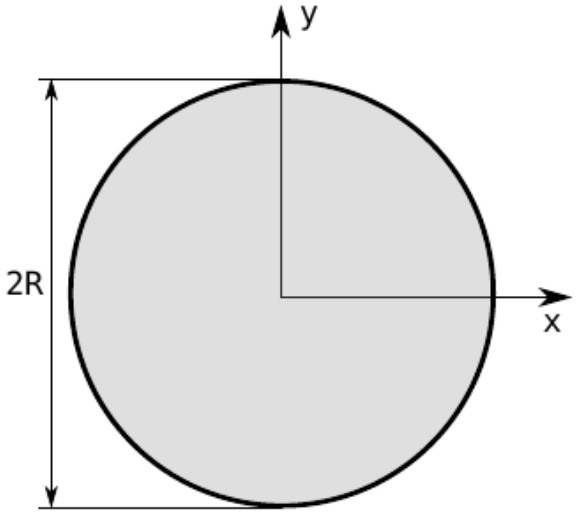


Coordinate systems: local

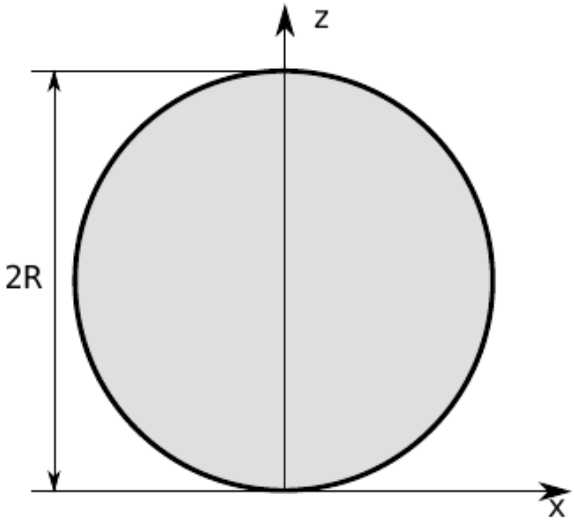
- Attached to each particle
- Rotations of particle are with respect to this local system
- Position of the particle is position of local origin in parent's coordinate system



Perspective



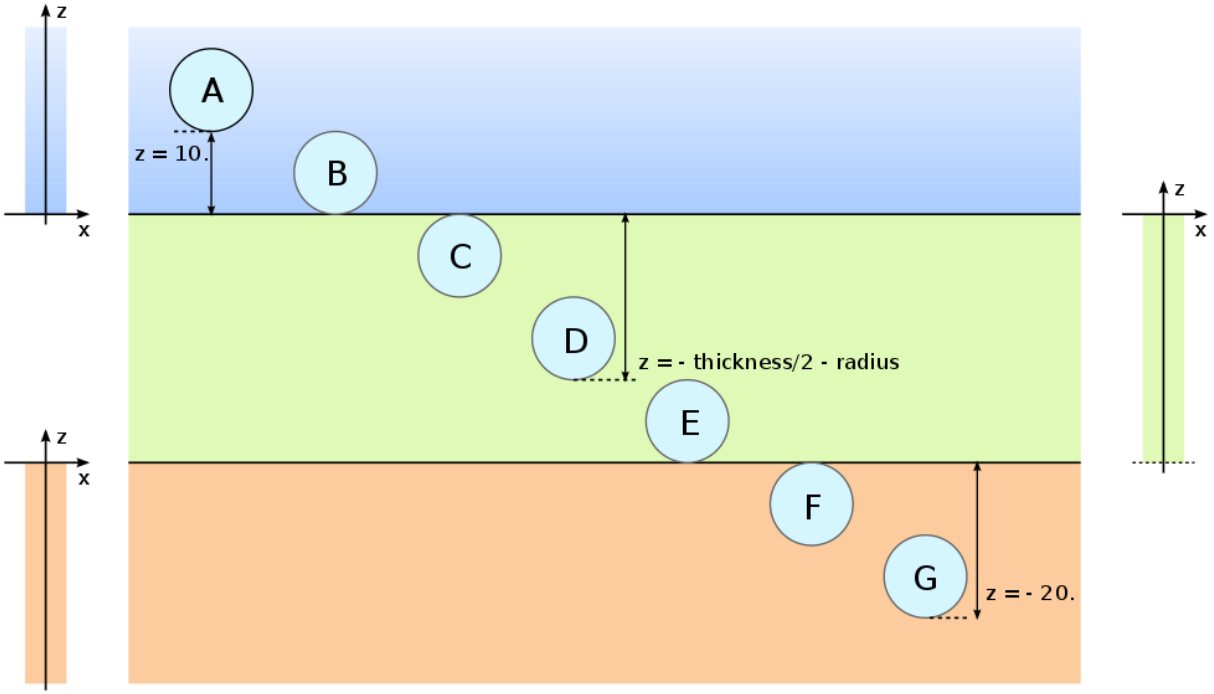
Top view



Side view

Coordinate systems: combined

- Particle's z-position is relative to layer's origin (top interface, except for top layer)
- This makes z negative for particles embedded in layers



BornAgain usage

- Intro
- Creating simulations:
  - Samples
  - Beam & detector
- Fitting:
  - Import real data
  - Run fit
- Python scripts
- Exercises