

# BornAgain Road Map

## Milestone: STAP meeting October 2020

Demonstrate new GUI (= preview of v2.0)

- mature views and controls, including
  - data-centric entry page
  - data loader
  - data plotter
    - different representations of reflectometry data
    - simultaneous plot of multiple data sets
  - sample model builder
    - repeated layer editor
    - SLD( $z$ ) visualization
- connection to BornAgain core in prototype state
- no fit interface yet

## Milestone v1.18 (Dec 2020)

- More stable and reliable algorithm for multilayer reflection & transmission computation
- Structural roughness in presence of magnetization

## Milestone v1.19 (May 2021)

- Python tutorial for fitting ToF data with proper resolution treatment; validated against experimental data
- Consolidated treatment of roughness, including slicing, without particles yet (*requires specification approved by Oct 2020*)

Python API overhaul

- Make some exported classes more "pythonic"
- More accurate form factor names

Prepare for GUI release

- Connection to BornAgain core finalized
- Decision taken on materials library (periodictable or equivalent C++ library)
- Fit interface designed

## Milestone v2.0 (Oct 2021)

New GUI for reflectometry

- Functionality as above, plus fitting
- Prototype off-specular viewer

## Milestone v2.1 (2022)

Full GUI support for

- Co-refinement
- polarized reflectometry
- off-specular scattering
- ToF fitting

# Comments on requirements from STAP meeting 2019

## Redesigned GUI for reflectometry

- Intuitive, data-centered GUI for reflectometry → *development started. Preview 9/20. Full simulation functionality 3/21. Release with fit support 9/21*

## Data loader

- Read different ASCII format (i.e. order of columns can vary, get input from ORSO) → *done in new GUI, can be demonstrated*
- Read ESS generated files (ask instrument scientists closer to hot commissioning) → *will be implemented as soon as format is defined*
- theta or 2theta as x-axis → *theta is default x axis in data loader*
- $R(Q^4)$  or  $R(Q^2)$  as y-axis → *supported through Python, can be implemented in GUI if need is affirmed*
- $R/R_f$  as y-axis → *done in new GUI, can be demonstrated*
- Read and use resolution column from file → *supported through Python, will be shown in GUI soon, full use depends on new fitting GUI*
- Scaling, calculation on data columns → *scale factor is implemented in new GUI*

## Plotting and Visualization

- Plot multiple datasets with simulations → *done in new GUI*
- Provide intuitive setup for publication quality graphs → *planned for 2021*
- Plot SLD profile (also with embedded particles) → *supported through Python, placeholder exists in new GUI, functionality to be attached in 2021*
- Plot parameter vs. figure-of-merit → *supported through Python, will be implemented after new fitting GUI*
- Calculate and show spin-asymmetry → *in Python 2021, in GUI 2022*

- Total reflection = unity (i.e. y axis is R not I) → *done through the scale factor mentioned above*

## Sample model

- Layer model editor → *Part of new GUI, see above*
- Repeating structures → *can be done through Python loop; can be set up from new GUI; connection with Core functionality to be implemented in autumn 2020*
- Layer roughness by sub-division → *needs theoretical clarification / user stories*

## Material parameterization through scattering-length density

- Material by atomic or molecular formula and density → *under Python, supported through Python module periodictable; for GUI support, we either need ways to call periodictable, or comparable functionality from a C/C++ library -- to be investigated*
- Database for materials (manual or online)/SLD calculator → *either comprised by the above, or user story needed*
- Include solvation (mixing of the layer SLD with the solvent SLD) → *the above should be generic enough to cover this request*
- Material by mixture of SLD (deuteration) → *ditto*
- Analytic function for SLD(z) → *done in new GUI*
- SLD calculator proteins → *beyond scope, requires contribution from collaboration partner*
- Wavelength dependent absorption (resonances) → *no need for action, is done correctly whenever there is an SLD distribution*

## Support for polarized reflectometry and magnetic scattering

→ *after milestone v1.18, we will reconsider these requests. More user input is needed.*

- non spin-flip can also be done with different SLDs not only matrix formalism.
- Magnetization surface density of sample
- Angle of magnetization to polarization
- Plot magnetic SLD profile
- Plot integral magnetic moment

## Instrument/physical effects

- Instrument specific import filters (templates/plugins) → *touches two different things*
  - *Data loader* → *see above*
  - *Metadata about instrument configuration* → *Waiting for instrument community to define standard*
- Proper resolution treatment for ToF (angular+WL) → *implemented in Python, validation planned for second half of 2020.*
- Simulate arbitrary q or 2theta points (e.g. ToF) → *user stories needed*

- Resonant scattering with user defined optical parameters, maybe from measurement → *beyond scope, requires contribution from interested scientists*

## Refinement

- Gradient descent refinement → *the BornAgain multfit wrapper comprises gradient descent; available in Python and in GUI*
  - log (weighing of data points) → *supported through Python and in GUI*
  - Gradient descent refinement  $\chi^2$  → *supported through Python and in GUI*
  - + errors → *see below (evaluation of parameter uncertainties)*
- Genetic/Global refinement → *the BornAgain multfit wrapper currently comprises 5 global optimizers, among them a genetic algorithm*
- User defined figure-of-merit → *supported through Python; GUI support is technically complicated, will be attempted if there is a strong use case.*
- Co-refinement multiple datasets → *supported through Python; GUI to be designed in after milestone v2.0, to be implemented in 2022; user stories needed*
  - different polarizations
  - different contrasts
  - Co-refinement x-ray/neutron
  - Same sample parameters for multiple probes (x-ray/neutron up/neutron down/...) and possible specification of few changing parameters (magnetization/deuteration per dataset)
  - Adjust weighting of co-refined datasets (e.g. x-ray vs. neutron)
- Parameter constraints → *supported through Python; GUI support to be decided later; user stories needed*
  - one parameter as function of others
  - Apply arbitrary mathematical constraints between variables within a model (over different contrasts)
- Evaluation of parameter uncertainties (from  $\chi^2$ ) → *implemented minimizers return parameter uncertainties, which however we found very unreliable. Deeper investigation is needed; expert help would be welcome.*

## Batch processing

*feasible through Python scripts*

## Bespoke models

*feasible through Python API*

## Scripting Interface

- Intuitive, pythonic interface → [R#2467](#), [R#2468](#), [R#2469](#), *scheduled for next API-breaking release, probably v2.0 in 2021*