

Fitting from Python

Gennady Pospelov
Jülich Centre for Neutron Science at MLZ

BornAgain school and user meeting
Garching, December 2018

day_2

Demo

Minimization and fitting in Python

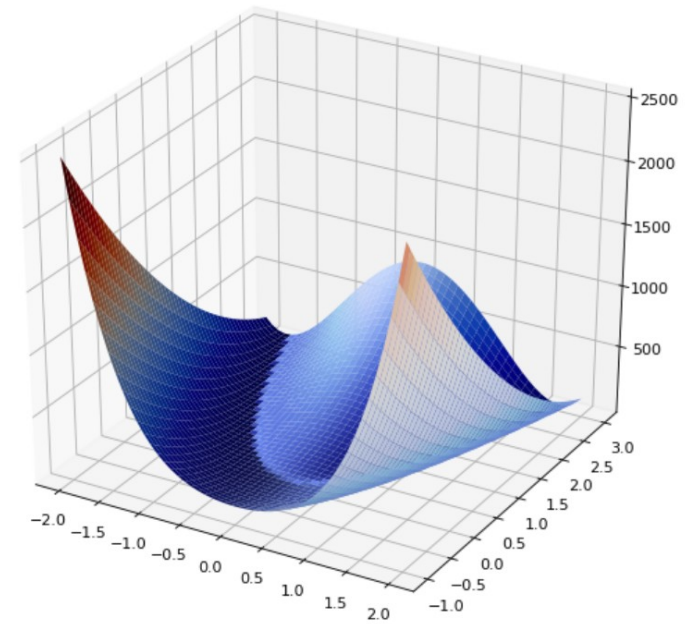
`pyapi06_rosenbrock_minimum.ipynb`

`pyapi06_lmfit_decaying_wave.ipynb`

Minimization example

Finding the minimum of multi argument function in given region

pyapi06_rosenbrock_minimum.ipynb



$$f(x, y) = (a - x)^2 + b(y - x^2)^2$$

```
In [87]: def rosenbrock(pars):  
    x = pars[0]  
    y = pars[1]  
    tmp1 = y - x * x  
    tmp2 = 1 - x  
    return 100 * tmp1 * tmp1 + tmp2 * tmp2
```

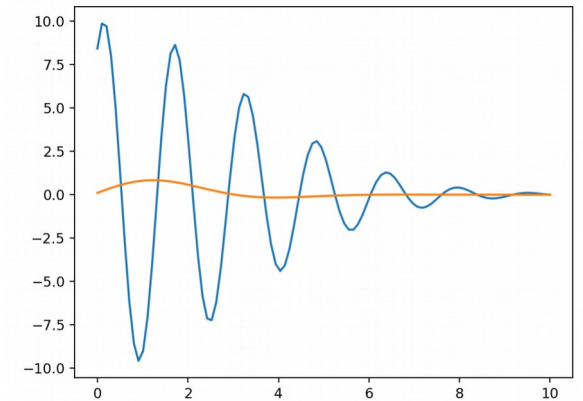
```
In [91]: x0 = np.array([-2.0, -1.0])  
res = minimize(rosenbrock, x0, method='nelder-mead', options={'xtol': 1e-8, 'disp': True})
```

```
Optimization terminated successfully.  
    Current function value: 0.000000  
    Iterations: 107  
    Function evaluations: 201
```

Fitting example

Finding the best set of parameter values to represent data according to some criteria

`pyapi06_lmfit_decaying_wave.ipynb`



```
In [ ]: import lmfit
```

```
In [ ]: def decaying_sin(params, x):  
    amp = params['amp']  
    phaseshift = params['phase']  
    freq = params['frequency']  
    decay = params['decay']  
    return amp * np.sin(x*freq + phaseshift) * np.exp(-x*x*decay)
```

```
In [ ]: def objective_function(params):  
    model = decaying_sin(params, x)  
    return (data - model) / eps
```

```
In [ ]: params = lmfit.Parameters()  
params.add('amp', value=1)  
params.add('decay', value=0.1)  
params.add('phase', value=0.1)  
params.add('frequency', value=1.0)
```

```
In [ ]: lmfit.minimize(objective_function, params, iter_cb=plotter)
```

Fitting in BornAgain from Python

Conceptually, is all about constructing an objective function that represents the difference between simulation and data, and passing it to a minimization engine

```
In [ ]: def create_simulation(params):  
        simulation = GISASSimulation()  
        ...  
        return simulation
```

```
In [ ]: def residual(params):  
        simulation = create_simulation(params)  
        simulation.runSimulation()  
        return data-simulation
```

```
In [ ]: params = {"cylinder_radius" : 5.0, "lattice_length" : 12.0}  
lmft.minimizer(residual, params)
```

New fitting API

SINCE RELEASE 1.13

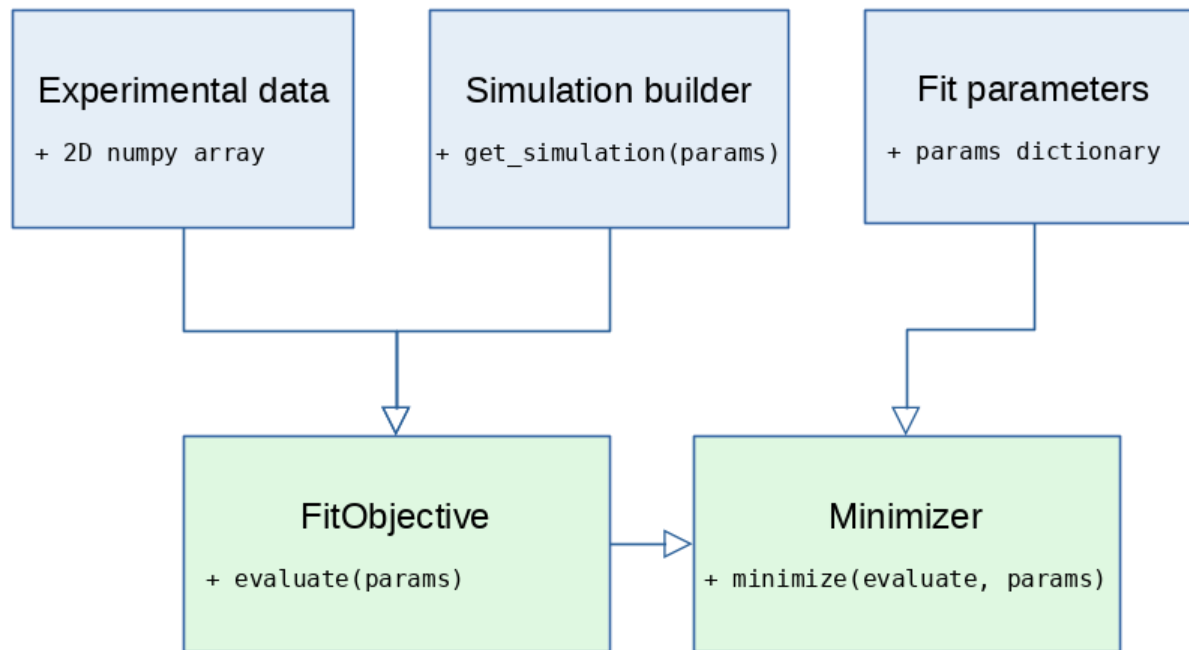
- Unified fitting interface
- Possibility to use external minimizers
- Custom objective functions
- Allow usage of our minimizer for optimization problems outside the BornAgain context.
- Allow the usage of other, possibly more advanced minimization libraries, for BornAgain fits.

Demo

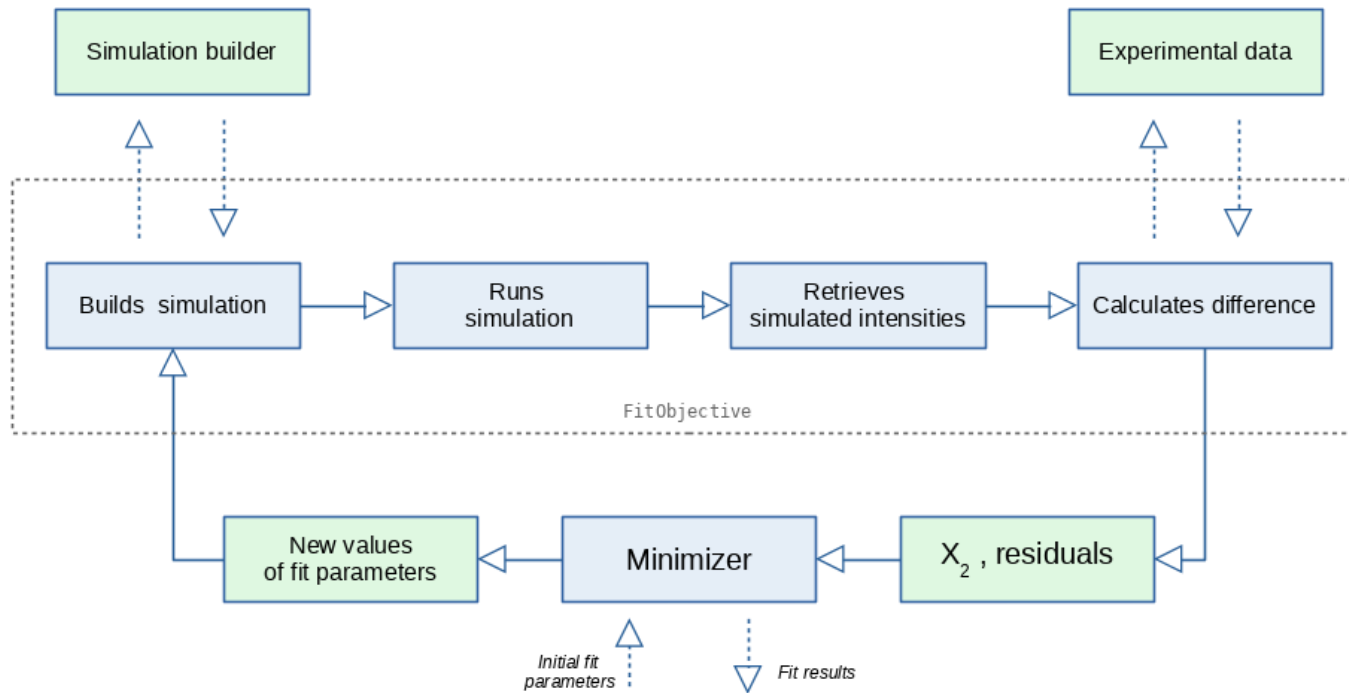
Exact Python equivalent of GUI fit from day_1

`pyapi07_hexspheres_fit.py`


Fitting components




Fitting workflow



Tutorials and fitting examples

 BornAgain

Home Gallery Download News Documentation Contact About 



Home > Documentation > Working with Python scripts > Fitting

Fitting

Fitting in BornAgain is used to find the values of sample or instrument parameters that best represent the data obtained in the user's scattering experiment.


In this section we will give a brief overview of the concept and describe how to setup and perform fits in BornAgain using the Python API. For a brief overview about fitting using the graphical user interface please see [here](#).



- [Introduction](#)
 - [Basic concept](#)
 - [Fitting in GISAS](#)
 - [Minimal fitting example](#)
- [Basic fitting tutorial](#)
- [Fitting highlights](#)
 - [Importing experimental data](#)
 - [Experiment description](#)
 - [Minimizer settings](#)


Documentation


- ⊕ Introduction
- ⊕ Getting started
- ⊕ Using Graphical User Interface
- ⊖ Working with Python scripts
 - Setup of a PyCharm project
 - ⊕ Basic simulation tutorial
 - Material types
 - ⊕ Detector types
 - Accessing simulation results
 - Particle positioning
 - Particle rotation
 - Particle composition
 - ⊕ Interference functions
 - Magnetic particles
 - Python API
 - ⊖ Fitting
 - ⊕ Introduction
 - Basic fitting tutorial
 - ⊕ Fitting highlights
 - ⊕ Sample model reference
 - ⊕ Getting help
 - ⊕ Developer's corner



Tutorials and fitting examples

 BornAgain

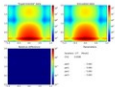
Home Gallery Download News Documentation Contact About 

Home > Documentation > Sample model reference > Fitting

Fitting

Basic examples

Minimal fit example

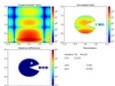


Basic fit example

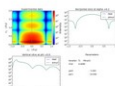
Minimizer settings

Consecutive fitting

Advanced examples




Fitting with masks



Fitting along slices

Find background

- [-] Documentation
 - [+] Introduction
 - [+] Getting started
 - [+] Using Graphical User Interface
 - [+] Working with Python scripts
 - [-] Sample model reference 
 - [+] Embedded particles
 - [+] Layered structures
 - [+] Interference functions
 - [+] Complex shapes
 - [+] Beam and detector
 - [+] Reflectometry
- [-] Fitting
 - [-] Basic examples
 - Minimal fit example
 - Basic fit example
 - Minimizer settings
 - Consecutive fitting
 - [-] Advanced examples
 - Fitting with masks
 - Fitting along slices
 - Find background
 - Simultaneous fit of two datasets
 - [-] Extended examples
 - Fitting reflectometry data
 - Custom objective function
 - Experiment at GALAXI
 - External minimizer
- [+] Miscellaneous
- [+] Getting help
- [+] Developer's corner