

# Documenting



# SasView

SasView  
**'documentation'**  
comes in many  
forms...



SasView 5.0.5

FileEditViewToolAnalysisFittingWindowHelp

Data Explorer

Data

Theory

Load data

Delete Data

Select all

> ☒ Ludox 1mm M4 TRANS\_SANS

Send data to

Fitting

☐ Batch mode

☐ Swap data

Plot

Create New

Append to

Graph1

Help

Documentation

Tutorial

Model Marketplace

Acknowledge

About

Welcome to SasView

Check for update

Optimizer: DREAM

Model name: sphere

Structure factor: hayter\_msa

Parameter	Value	Error	Min	Max	Units
<input type="checkbox"/> scale	1		0.0	$\infty$	
<input checked="" type="checkbox"/> background	0.86891	0.0042623	$-\infty$	$\infty$	$\text{cm}^{-1}$
structure_factor_mode	P*5				
radius_effective_mode	radius				
sphere					
<input type="checkbox"/> sld	3		$-\infty$	$\infty$	$10^{-6}/\text{\AA}^2$
<input type="checkbox"/> sld_solvent	6.3		$-\infty$	$\infty$	$10^{-6}/\text{\AA}^2$
<input checked="" type="checkbox"/> radius	158.27	0.11678	0.0	$\infty$	$\text{\AA}$
Polydispersity					
Distribution	0.10374	0.00043217			lognormal
hayter_msa					
<input type="checkbox"/> radius_effective	159.12		0.0	$\infty$	$\text{\AA}$
<input checked="" type="checkbox"/> volfraction	0.46723	0.00067104	0.0	0.74	None
<input checked="" type="checkbox"/> charge	3.6438	0.23768	$1\text{e-}06$	200.0	e
<input type="checkbox"/> temperature	298		0.0	450.0	K
<input type="checkbox"/> concentration_salt	0.0		0.0	$\infty$	M
<input type="checkbox"/> dielectconst	78		$-\infty$	$\infty$	None

Options

☒ Polydispersity

☐ Magnetism

Fitting details

Min range 0.00714  $\text{\AA}^{-1}$

Max range 0.25169  $\text{\AA}^{-1}$

Smearing: Use dQ Data

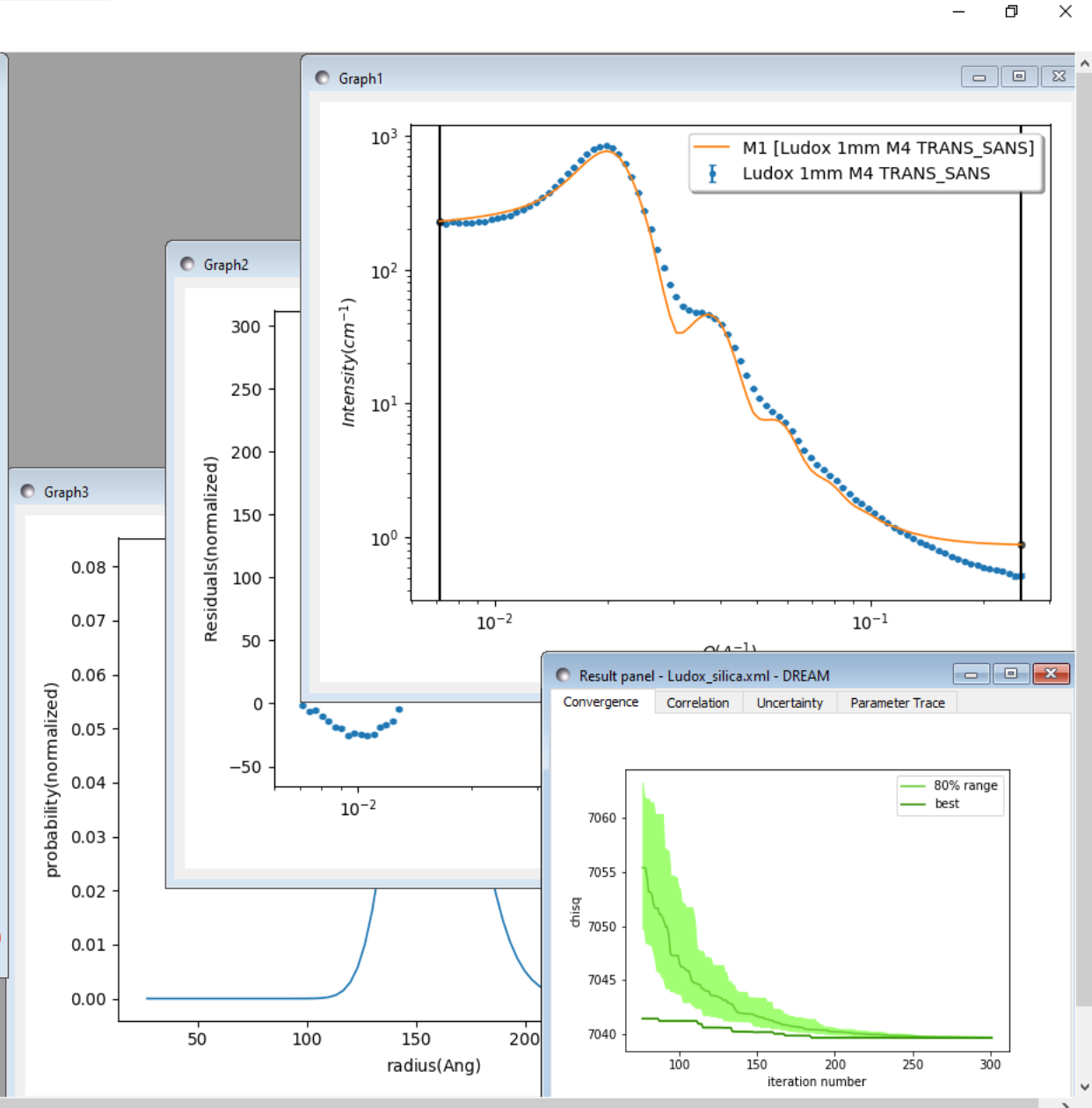
Fitting error

$\chi^2$  6657.1

Compute/Plot

Fit

Help



Log Explorer

M1.background: 0.869

M1.radius.width: 0.104

22:15:07 - INFO: 2022-06-09 22:15:07 == Steps: 301 of 300 chisq: 7.04e+03 ETA: 0s

M1.background: 0.869

M1.radius.width: 0.104

M1.volfraction: 0.467

M1.volfraction: 0.467

M1.radius: 158

M1.radius: 158

Next topic  
SasView User Documentation

This Page  
Show Source

Quick search

Go

# SasView Documentation Index

- USER DOCUMENTATION
- DEVELOPER DOCUMENTATION
- Release Notes
- Features
- Downloading and Installing
- Known Issues
- SasView Website
- Frequently Asked Questions

## USER DOCUMENTATION:

- ★★ • Fitting model descriptions ★★
- How to use SasView
- ‘Encyclopedia content’

## DEVELOPER DOCUMENTATION:

- Is mostly auto-generated
- Lists of program modules/classes
- Overview of Sasmodels

Available in-program and online – but single-source  
C:/SasView-5.0.5/doc/index.html & <https://www.sasview.org/docs/index.html>

## USER DOCUMENTATION:

- Is written in ReStructured text (.rst)
- Lightweight, portable, human-readable, but also software-processable
- Supports LaTeX-like math

```
Invariant
^^^^^^^^
```

```
SasView implements the invariant calculation for a two-phase (or pseudo
two-phase) system, which represents the most commonly encountered situation.
The invariant for this is
```

```
.. math::
```

$$Q^* = \{2 \pi^2 (\Delta\rho)^2 \phi_1 \phi_2\}$$

```
where  $\Delta\rho = (\rho_1 - \rho_2)$  is the SLD contrast and  $\phi_1$  and
 $\phi_2$  are the volume fractions of the two phases ( $\phi_1 + \phi_2 = 1$ ).
Thus from the invariant one can either calculate the volume fractions of the
two phases given the contrast or, calculate the contrast given the volume
fraction. However, the current implementation in SasView only allows for the
former: extracting the volume fraction given a known contrast factor.
```

invariant\_help.rst

Keep lines to  
≤80 characters

Do not end  
lines with  
whitespace!  
It upsets  
'linters'

We then use  **SPHINX**  
PYTHON DOCUMENTATION GENERATOR

to autoconvert the .rst to the desired output (.html, .pdf, ...)



[Previous topic](#)[raspberry](#)[Next topic](#)[spherical\\_sld](#)[This Page](#)[Show Source](#)[Quick search](#)[Go](#)

# sphere

Spheres with uniform scattering length density

Parameter	Description	Units	Default value
scale	Scale factor or Volume fraction	None	1
background	Source background	cm <sup>-1</sup>	0.001
sld	Layer scattering length density	10 <sup>-6</sup> Å <sup>-2</sup>	1
sld_solvent	Solvent scattering length density	10 <sup>-6</sup> Å <sup>-2</sup>	6
radius	Sphere radius	Å	50

The returned value is scaled to units of cm<sup>-1</sup> sr<sup>-1</sup>, absolute scale.

For information about polarised and magnetic scattering, see the [Polarisation/Magnetic Scattering documentation](#).

## Definition

The 1D scattering intensity is calculated in the following way (Guinier, 1955)

$$I(q) = \frac{\text{scale}}{V} \cdot \left[ 3V(\Delta\rho) \cdot \frac{\sin(qr) - qr \cos(qr)}{(qr)^3} \right]^2 + \text{background}$$

where *scale* is a volume fraction, *V* is the volume of the scatterer, *r* is the radius of the sphere and *background* is the background level. *sld* and *sld\_solvent* are the scattering length densities (SLDs) of the scatterer and the solvent respectively, whose difference is  $\Delta\rho$ .

Note that if your data is in absolute scale, the *scale* should represent the volume fraction (which is unitless) if you have a good fit. If not, it should represent the volume fraction times a factor (by which your data might need to be rescaled).

The 2D scattering intensity is the same as above, regardless of the orientation of  $\vec{q}$ .

## Validation

Validation of our code was done by comparing the output of the 1D model to the output of the software provided by the NIST (Kline, 2006).

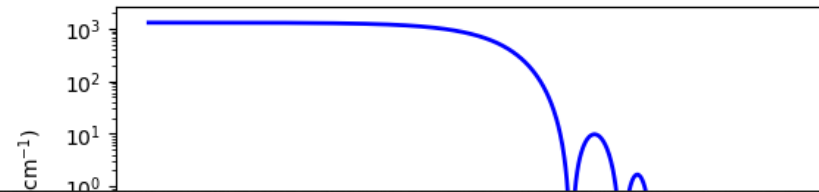


Figure autogenerated from the default model parameters

For information about polarised and magnetic scattering, see the [ref:'magnetism' documentation](#).

## Definition

The 1D scattering intensity is calculated in the following way (Guinier, 1955)

.. math::

$$I(q) = \frac{\text{scale}}{V} \cdot \left[ 3V(\Delta\rho) \cdot \frac{\sin(qr) - qr \cos(qr)}{(qr)^3} \right]^2 + \text{background}$$

where *scale* is a volume fraction, *V* is the volume of the scatterer, *r* is the radius of the sphere and *background* is the background level. *sld* and *sld\_solvent* are the scattering length densities (SLDs) of the scatterer and the solvent respectively, whose difference is  $\Delta\rho$ .

Note that if your data is in absolute scale, the *scale* should represent the volume fraction (which is unitless) if you have a good fit. If not, it should represent the volume fraction times a factor (by which your data might need to be rescaled).

The 2D scattering intensity is the same as above, regardless of the orientation of  $\vec{q}$ .

## Validation

Validation of our code was done by comparing the output of the 1D model to the output of the software provided by the NIST (Kline, 2006).

## References

1. A Guinier and G. Fournet, \*Small Angle Scattering of X-Rays\*, John Wiley and Sons, New York

## Authorship and Verification

\*\*\*Author:\*\*\*  
 \*\*\*Last Modified by:\*\*\*  
 \*\*\*Last Reviewed by:\*\*\* S King and P Parker \*\*\*Date:\*\*\* 2013/09/09 and 2014/01/06  
 \*\*\*

```
import numpy as np
from numpy import inf
```

```
name = "sphere"
title = "Spheres with uniform scattering length density"
```

```
description = """
```

```
P(q)=(scale/V)*[3V(sld-sld_solvent)*(sin(qr)-qr*cos(qr))
```

```
/ (qr)^3]^2 + background
```

```
r: radius of sphere
```

```
V: The volume of the scatter
```

```
sld: the SLD of the sphere
```

```
sld_solvent: the SLD of the solvent
```

```
***
```

```
category = "shape:sphere"
```

```
# ["name", "units", default, [lower, upper], "type", "description"],
```

```
parameters = [{"sld", "1e-6/Å²", 1, [-inf, inf], "sld",
```

```
"Layer scattering length density"],
```

```
["sld_solvent", "1e-6/Å²", 6, [-inf, inf], "sld",
```

```
"Solvent scattering length density"],
```

```
["radius", "Å", 50, [0, inf], "volume",
```

```
"Sphere radius"],
```

```
]

```

```
source = ["lib/sas_3jlx_x.c", "sphere.c"]
```

```
have_Pq = True
```

```
radius_effective_modes = ["radius"]
```

```
#single = False
```

sphere.py

## BUILDING THE DOCUMENTATION:

- The *full* documentation comprises that from *Sasmodels* plus that from *SasView*, so **if you want the model descriptions you must build the *Sasmodels* documentation first!**

```
> cd <sasmodels_installation_folder>\doc  
> make clean html
```


then

```
> cd <sasview_installation_folder>\doc  
> python setup.py build install  
> python setup.py build docs
```

(only if you have not already built SasView)


# WRITTEN TUTORIALS:

- See <https://github.com/SasView/sasview/wiki/TutorialsTNGForAuthors>
- Are written in  LibreOffice Writer (download from <https://www.libreoffice.org/>)
- Are saved in OpenDocument Text Format (.fodt)
- The .fodt can be processed by  SPHINX PYTHON DOCUMENTATION GENERATOR *(but is not being at present)*



ABOUT ▼LINKS & DOWNLOADSDOCUMENTATIONCONTENT ▼HELP ▼

Go to each FitPage in turn and select the core shell sphere model. The three theory curves will be added to the graph.



Enter the following SLD values in the appropriate FitPage's:

Constant (SLD)	SLD core	SLD shell	SLD solvent
core	-0.00017 Å <sup>-2</sup>	-0.00017 Å <sup>-2</sup>	-0.00017 Å <sup>-2</sup>
shell	-0.00017 Å <sup>-2</sup>	-0.00017 Å <sup>-2</sup>	-0.00017 Å <sup>-2</sup>
solvent	-0.00017 Å <sup>-2</sup>	-0.00017 Å <sup>-2</sup>	-0.00017 Å <sup>-2</sup>

The objective of this simultaneous fit will be to find optimal values for the radius (of the shell) and thickness of the spherical core. The radius is the only parameter that is shared between the three datasets. The thickness is the only parameter that is shared between the two shell datasets.

We now need to tell SasView that the radius for the 'core' dataset is the same as the radius for the 'shell' dataset. We do this by clicking on the 'link' button in the 'core' dataset, and we are on 'SLD'. This is called 'linking' datasets. But you can only link parameters that are not in 'constant'. So in each FitPage, check the radius and thickness parameters. However, we also need to optimize the scale and background parameters in each FitPage.

**Tip: Always check the parameters you want to constrain in their respective FitPage's before entering the Simultaneous Fit page.**

Now go to the Menu Bar and click **Fit** followed by **Simultaneous Fit**.

0014

In the Constant & Shell Fit page, check the boxes under Model Title for just **Sphere** to select from theories that you want to constrain constants for. For this example, check all three theories. Then, in the section of the page called Fit Constraints, check the **SLD** radio button to add Constraints.

To constrain all identically named parameters to be simultaneously to the same value across all the selected theories we can use the Easy Setting drag-and-drop feature. There are, however, several steps that we can set up the constraint equations. Here we shall use the 'core' constant (SLD) as the reference. So set **SLD core** and click **SLD core**. Then set **SLD shell** and click **SLD core**.

Now go to the Menu Bar and click **Fit** followed by **Simultaneous Fit**.

**Tip: If you need to scale parameter values between FitPage's then use the free-hand operators on the Scale Setting. The right hand side of the equality can be of the form: **SLD parameter** / **SLD parameter**.**

As we are assuming the volume fraction of deuterium in each sample was different, remove the 'core' scale constraints, and because the different datasets represent samples containing different amounts of D<sub>2</sub>O and therefore have different incoherent scattering contributions, also remove the 'core' background constraints.

But as it is the 'shell' constant (SLD) which should provide the most sensitivity to the thickness parameter, we need to change the constant equations leaving with thickness in 'core'.

M1 thickness = M1 thickness  
M2 thickness = M1 thickness

So we now have constraints that look like this:

0014

## Tutorials:

- [Old SasView tutorial](#) (PDF) – dated but still useful
- “Getting started with SasView” (PDF) for [versions 3.x/4.x](#) & [versions 5.x](#)
- “Basic 1D Fitting in SasView” (PDF) for [versions 3.x/4.x](#) & [versions 5.x](#)
- “Simultaneous 1D Fitting in SasView” (PDF) for [versions 3.x/4.x](#) & [versions 5.x](#)
- “P(r) Inversion Analysis in SasView” (PDF) for [versions 3.x/4.x](#) & [versions 5.x](#)
- “Correlation Function Analysis in SasView” (PDF) for [versions 4.x](#) & [versions 5.x](#)
- “Subtracting a Model Calculation from Data in SasView” (PDF) for [versions 4.x](#) & [versions 5.x](#)

Lots of scope for community contributions here!



# VIDEO TUTORIALS:

- See <https://www.youtube.com/channel/UCxvD3ysXJ05l6MgY7YKjEFQ> or <https://www.sasview.org/documentation/>

The screenshot displays the YouTube channel page for SasView. The channel has 116 subscribers and a 'SUBSCRIBE' button. The video uploads section shows a grid of videos with titles, view counts, and upload dates. The videos include:


- SasView 5.0.5 - How to use the new scattering calculator...** (122 views • 5 months ago)
- SasView - A 'Swiss Army Knife' for SAS Data Analysis** (481 views • 8 months ago)
- Scattering Length Density Calculator in SasView** (584 views • 3 years ago)
- Using the P(r) calculator in SasView** (962 views • 3 years ago)
- Using the Correlation Function Analysis...** (582 views • 3 years ago)
- Introduction to applying the beta approximation in...** (302 views • 3 years ago)
- Calculating the Scattering Invariant in SasView** (419 views • 3 years ago)

The background of the channel banner features the SasView logo and a collage of various scientific plots and data visualizations.

Lots of scope for community contributions here too!

# WEBSITE:

- See <https://www.sasview.org/> < <https://github.com/SasView/sasview.github.io>



**SasView**  
SasView for Small Angle Scattering Analysis

**Download The Latest Release Version of SasView**

Version 4	Version 5
<a href="#">Download Version 4.2.2</a>	<a href="#">Download Version 5.0.5</a>
Released on May 20, 2019. <a href="#">Important Message!</a>	Released on June 3, 2022. <a href="#">See what's new!</a>

**A SAS Community Project launched from the NSF DANSE effort**  
SasView is a Small Angle Scattering Analysis Software Package, originally developed as part of the NSF DANSE project under the name SansView, now managed by an international collaboration of facilities. Feedback and contributions are welcome and encouraged.

**News:**

**Version 5.0.5**  
Posted on June 3, 2022  
Version 5.0.5 was released on Jun 03, 2022! This is a point release which fixes some issues reported in earlier versions of 5.0.x. A few highlights are: [\[Read More\]](#)

## FAQ

Here are the answers to some common questions about SasView

[What is SasView?](#)

[Is there a SasView Manual?](#)

[What platforms does SasView run on?](#)

[Do I need to install Python or C or any compilers before I install SasView?](#)

[What do xcrun errors mean?](#)

[Can SasView make use of my GPUs?](#)

[Can I stop SasView trying to use my GPUs?](#)

[Can I use SasView to analyse SANS or USANS data?](#)

[Can I use SasView to analyse SAXS or USAXS data?](#)

[Can SasView be used in a commercial environment?](#)

[What format should my data be in?](#)

[What units should my data be in?](#)

[Why is SasView no longer loading my data?](#)

[Does SasView fit data in its I vs Q representation, or as Log I vs Log Q?](#)

[Many models have a scale parameter; what is this?](#)

[How can I put my data on an absolute intensity scale?](#)

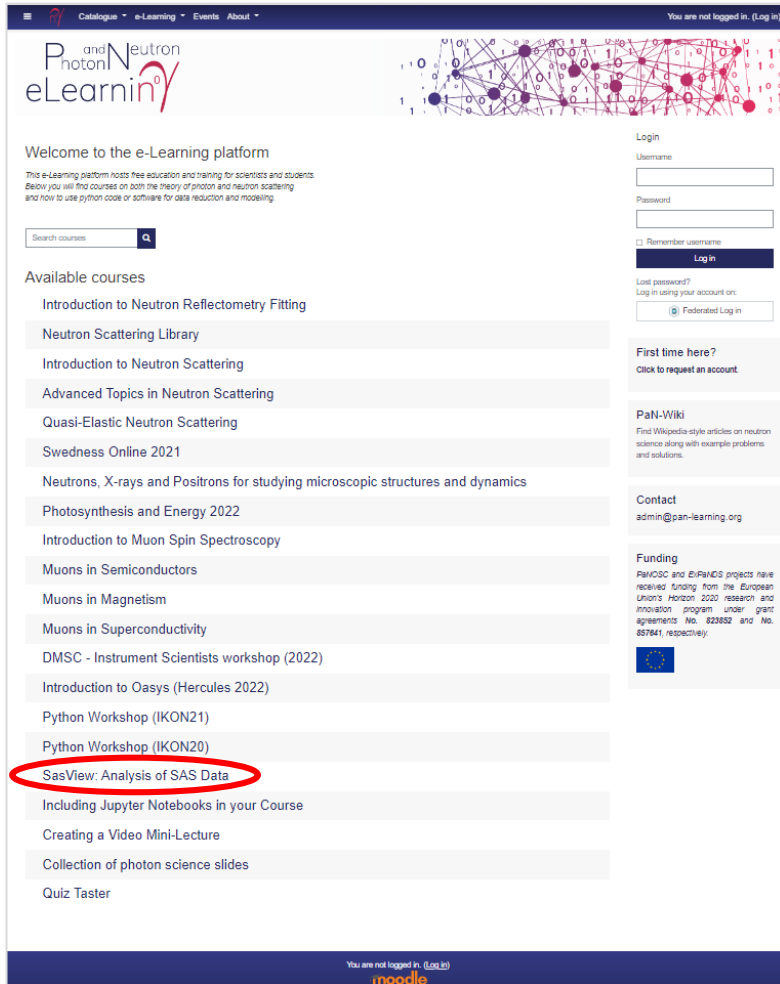
[Why do I get non-sensical parameter values if I try to simultaneously model-fit the scale factor and the scattering length densities?](#)

[What value should I use to specify polydispersity?](#)

[Is the Schulz\(-Zimm\) polydispersity function used in SasView a number-average or a](#)

# OTHER ONLINE RESOURCES:

- See <https://pan-learning.org/>
- Contains a 'SasView Analysis' module
- Written in Moodle



The screenshot shows the homepage of the Pan-Learning Moodle platform. The header includes navigation links for Catalogue, e-Learning, Events, and About, along with a login status indicator. The main content area is divided into two columns. The left column, titled 'Available courses', lists various modules, with 'SasView: Analysis of SAS Data' highlighted by a red circle. The right column contains a login form, a 'First time here?' link, a 'PaN-Wiki' section, a 'Contact' section, and a 'Funding' section. The footer displays the Moodle logo and a login status indicator.

Navigation: Catalogue e-Learning Events About

You are not logged in. (Log in)

Phonon and Neutron eLearning

Welcome to the e-Learning platform

This e-Learning platform hosts free education and training for scientists and students. Below you will find courses on both the theory of phonon and neutron scattering and how to use python code or software for data reduction and modelling.

Search courses

Available courses

- Introduction to Neutron Reflectometry Fitting
- Neutron Scattering Library
- Introduction to Neutron Scattering
- Advanced Topics in Neutron Scattering
- Quasi-Elastic Neutron Scattering
- Swedness Online 2021
- Neutrons, X-rays and Positrons for studying microscopic structures and dynamics
- Photosynthesis and Energy 2022
- Introduction to Muon Spin Spectroscopy
- Muons in Semiconductors
- Muons in Magnetism
- Muons in Superconductivity
- DMSC - Instrument Scientists workshop (2022)
- Introduction to Oasys (Hercules 2022)
- Python Workshop (IKON21)
- Python Workshop (IKON20)
- SasView: Analysis of SAS Data**
- Including Jupyter Notebooks in your Course
- Creating a Video Mini-Lecture
- Collection of photon science slides
- Quiz Taster

Login

Username

Password

☐ Remember username

Login

Lost password?

Log in using your account on:

☐ Federated Log in

First time here?

Click to request an account.

PaN-Wiki

Find Wikipedia style articles on neutron science along with example problems and solutions.

Contact

admin@pan-learning.org

Funding

PaN-LEARN and e-LEARN projects have received funding from the European Union's Horizon 2020 research and innovation program under grant agreements No. 823652 and No. 857841, respectively.

You are not logged in. (Log in)

moodle



