Absorption-Correction Factors for Scattering from Flat or Tubular Samples: Open-Source Implementation libabsco, and Why it Should be Used with Caution.

Joachim Wuttke*
Forschungszentrum Jülich GmbH,
Jülich Centre for Neutron Science at FRM II,
Lichtenbergstraße 1, 85747 Garching, Germany
(Dated: draft, May 30, 2012)

In the analysis of scattering experiments, correction factors are used to compensate for self-absorption in the sample and for absorption in the sample container. This short note documents the open-source library `libabsco` that provides computations of absorption corrections for flat or tubular samples. At this occasion, it is pointed out that the use of absorption corrections for multiple-scattering losses is inconsistent, and that there are better alternatives to the subtraction of container scattering.

I. INTRODUCTION

In X-ray or neutron scattering, it is well-established routine that recorded intensities be corrected for attenuation. When attenuation takes place in both the sample and its container, then for each scattering angle three correction factors are needed. The open-source library `libabsco` provides functions for the computation of these factors. It is written in the variant C99 of the programming language C. Source code, documentation, and build utilities are available at `http://apps.jcns.fz-juelich.de/absco`.

The computations are neither involved nor new; I publish this software and its documentation with no other pretension than to provide clean and clear code that will save colleagues from reimplementing these computations again and again. Specifically, this work responds to requests from users of high-resolution inelastic neutron spectrometers. Should it be found useful in other fields, I would be grateful for feedback. Bug reports, feature requests, corrections, and extensions will also be welcome.

II. RESERVATIONS

Before explaining how the library works, I shall enunciate three reservations against overtrusting straightforward self-absorption corrections.

A. Multiple scattering is not absorption

In neutron scattering, the absorption cross section is often much smaller than the scattering cross section. In this case, the attenuation of the incident and the scattered ray is dominated by multiple scattering, not by absorption. Inserting a total cross section, dominated by scattering, into a correction formula derived for pure absorption is dangerously inconsistent: one compensates for the *losses* but not for the *gains* caused by multiple scattering though both are of the same magnitude.

It might be preferable to proceed directly from a first-glance data analysis, with no multiple-scattering corrections at all, to an accurate analysis that incorporates a full-fledged multiple-scattering simulation, without spending time on intermediate corrections that do not hold what they promise.

B. Avoid data subtraction

Typically, absorption corrections are used in a *data reduction* procedure: weigh the measured scattering function with a correction factor, and subtract the measured container scattering weighed with another factor, to obtain a better approximation to the true scattering function. While this is the most anschaulich way of data analyis, and very often the only practicable one, it is unsatisfactory because mathematically unjustified. This becomes obvious when the container subtraction leads to negative counts; there is no way then to properly weigh data points in fitting.

For a statistically meaningful data analysis it is better to avoid invasive data-reduction steps and to fit data that are as close as possible to the measured raw counts. As a radical alternative to the *data reduction* paradigma one may dream of a *virtual instrument* software that performs a full simulation of the scattering experiment within a Bayesian determination of the sample's scattering function. At least one should avoid subtraction steps, and fit the experimental scattering law with a theoretical model that comprises the container scattering. In such a procedure, absorption correction factors may still be needed for proper weighing.

C. Be realistic about geometries

Samples for neutron scattering often consist of a thin layer of sample material encapsulated in a thin metallic

*Electronic address: `j.wuttke@fz-juelich.de`

container. Since the container is prone to deformation by fabrication-induced stress and thermal dilations, there is no guarantee that the sample layer is homogenous. In the worst case, the soft matter under study starts to flow during the experiment and congregates at the bottom of the container.

Therefore, experienced experimentalists tend to use the sample thickness as an adjustable parameter that can be hand-tuned to ensure a plausible outcome of the self-absorption correction procedure. This is one more reason to abandon the idea of data reduction altogether and to integrate the adjustment of geometry parameters into an overall fitting procedure.

III. FOUNDATIONS

This section documents the basic definitions and assumptations that underlie the computation of absorption factors in of `libabsco`. Most of this information can also be found in the manual page `absco(7)` that comes with the source distribution.

A. Absorption factors

Using established notation [1, 2], we describe the effect of self absorption as

$$M_S = A_{S,S} I_S \tag{1}$$

where I_S is the scattering from sample S in absence of self-absorption (*ideal scattering*), M_S is the scattering from sample S in presence of self-absorption (*measured scattering*), and $A_{S,S}$ is the self-absorption factor.

More generally, an absorption factor $A_{X,Y}$ must be applied if the scattering event takes place in X, while absorption occurs in all of Y, where Y must be a superset of X. Usually, a sample S is contained in a container C, and scattering by the empty container is measured separately:

$$M_C = A_{C,C} I_C, \tag{2}$$

$$M_{SC} = A_{C,SC} I_C + A_{S,SC} I_S. \tag{3}$$

In consequence, the ideal scattering of the sample can be calculated from M_{SC} and M_C:

$$I_S = \frac{1}{A_{S,SC}} M_{SC} - \frac{A_{C,SC}}{A_{S,SC} A_{C,C}} M_C. \tag{4}$$

`libabsco` provides functions to compute $A_{S,SC}$, $A_{C,SC}$ and $A_{C,C}$ for different sample geometries.

Absorption factors for given scattering angles Θ are computed as

$$A_{X,Y}(\Theta) = \left\langle \exp \sum_{Z \in Y} -\mu_{\mathrm{i}}^Z p_{\mathrm{i}}^Z(\boldsymbol{r}) - \mu_{\mathrm{f}}^Z p_{\mathrm{f}}^Z(\boldsymbol{r}, \Theta) \right\rangle_X . \tag{5}$$

The sum runs over different materials Z, in our application maximally over C and S. The attenuation coefficient $\mu_{\mathrm{i,f}}^Z$ for incident and final rays may differ in case of inelastic scattering. The function $p_{\mathrm{i}}^Z(\boldsymbol{r})$ is the length of the incident path towards $\boldsymbol{r}$ across Z. The outgoing path length $p_{\mathrm{f}}^Z(\boldsymbol{r}, \Theta)$ depends also on the scattering angle; it comprises $p_{\mathrm{i}}^Z(\boldsymbol{r}) = p_{\mathrm{f}}^Z(\boldsymbol{r}, \pi)$ as a special case,

The brackets in (5) designate an average over scattering locations $\boldsymbol{r} \in X$:

$$\langle f(\boldsymbol{r}) \rangle_X = \frac{\int_X \mathrm{d}^3 r\, f(\boldsymbol{r})}{\mathrm{Vol}(X)}. \tag{6}$$

Except for the simplest geometry, the integral must be computed numerically. If the integral is multi-dimensional, the Monte-Carlo method comes into mind, but in order to keep data-reduction procedures fully reproducible, only deterministic methods are used.

B. Transmission, backscattering factor

In a backscattering spectrometer, scattered neutrons traverse the sample a second time on their way from the analyzer to the detector. This is expressed by f_{BS}, the fraction of backscattered neutrons, which should be 1 for exact backscattering and 0 for all other situations.

It is assumed that the impact points of the reentrant neutrons are uniformly distributed over the sample section, and that they uncorrelated with position and angle of the scattering event. The backscattering attenuation factor is then $(1 - f_{\mathrm{BS}}) + f_{\mathrm{BS}} T(\pi - \Theta, \mu_{\mathrm{f}})$ where

$$T(\chi, \mu) = \left\langle \exp \sum_{Z \in Y} -\mu^Z p_{\mathrm{f}}^Z(\boldsymbol{r}, \chi) \right\rangle_{\mathrm{section}(\chi)} \tag{7}$$

is the transmission for propagation direction χ.

The standard transmission is $T(0, \mu_{\mathrm{i}})$. The average in (7) runs over the sample section as seen from direction $\pi - \chi$. Therefore T must be distinguished from the collimated-beam transmission that is routinely measured at some instruments [3].

To evaluate (7) in a first approximation, one could use

$$\langle \exp -\mu p \rangle \doteq \exp -\mu \langle p \rangle. \tag{8}$$

However, this approximation is not made in `libabsco`; when necessary, the average in (7) is computed by numeric integration. Fig. 1 demonstrates the inaccuracy of (8) for a tubular sample.

C. Units

Lengths and attenuation coefficients form dimensionless products. Therefore, the functions in `libabsco` could be left agnostic about units; it is in the users' responsability to provide attenuation coefficients μ in units that are reciprocal to those used for lengths.

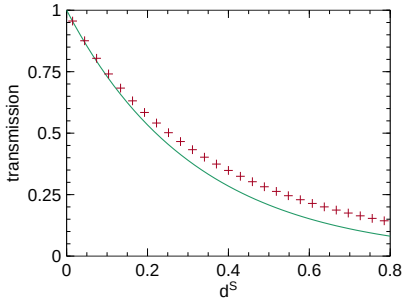

FIG. 1: Transmission of a containerless tubular sample with radius $R_o = 30$ as function of the sample layer thickness d^S; implicit length unit set by the attenuation coefficient $\mu^S = 1$. Symbols are from numeric integration, the line shows approximation (8) with $\langle p \rangle = \pi d^S$.

Input angles are understood to be given in degrees; for internal use they are converted to radians.

IV. FUNCTION CALLS

This section duplicates the description of the application programming interface (API) from the manual page `absco(7)`.

A. Common interface

The API is declared in the header

```
#include <absco.h>
```

For each geometry, there is one function that computes the absorption coefficients. The function call starts with eight common arguments, followed by geometry-specific arguments:

```
void absco_<geometry> (
  double *A_S_SC,              A_{S,SC}
  double *A_C_SC,              A_{C,SC}
  double *A_C_C,               A_{C,C}
  double scattering_angle,     Θ
  double mu_i_S,               μ^S_i
  double mu_f_S,               μ^S_f
  double mu_i_C,               μ^C_i
  double mu_f_C,               μ^C_f
  <geometry parameters>)
```

For brevity, `const` qualifiers are not listed here.

The three pointer arguments carry the return values. To compute only $A_{S,S}$, one must supply `A_C_SC=A_C_C=NULL` and `mu_i_C=mu_f_C=0`. All calculations involving the sample container are then skipped.

Starting with the scattering angle Θ, all remaining arguments are input parameters. If an input parameter is found to be invalid, an error message is written to `stderr` and the program `exit`s with an appropriate `errno`.

Another function returns the transmission $T(\chi, \mu^{S,C})$:

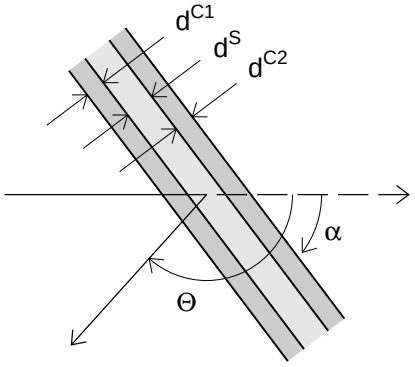

FIG. 2: Flat-slab geometry.

```
double transmission_<geometry> (
  [double outgoing_angle,]     χ
  double mu_S,                 μ^S
  double mu_C,                 μ^C
  <geometry parameters>)
```

As explained above, the transmission of the incident beam is $T(0, \mu_\mathrm{i}^{S,C})$; to compute the attenuation of backscattered neutrons, one needs $T(\pi - \Theta, \mu_\mathrm{f}^{S,C})$. If the sample has cylindrical symmetry, the transmission does not depend on the outgoing angle χ, and this argument is dropped.

B. Flat samples

The simplest sample geometry is a flat slab. The orientation of the slab is expressed by an angle α as shown in Fig. 2. For $\alpha = 90°$, the slab is perpendicular to the incident beam. In experiments, frequent choices are $\alpha = 45°$ (most scattering angles are in reflection) and $\alpha = 135°$ (most scattering angles in transmission). For computing absorption factors, infinite lateral extension is assumed, which is a good approximation except for scattering angles $\Theta \simeq \alpha$, in which case self absorption is so strong that no correction whatsoever will restore a reliable scattering signal.

The *geometry parameters* in the API specification of `absco_slab` and `transmission_slab` are

```
double slab_angle,           α
double thickness_S,          d^S
double thickness_C_front,    d^C1
double thickness_C_rear      d^C2
```

A typical curve $A_{S,S}(\Theta)$, assuming no container scattering and no backscattering, is shown in Fig. 3.

C. Tubular samples

In tubalar geometry, the sample layer (thickness d^S, outer radius R_o^S) is enclosed between an inner and an

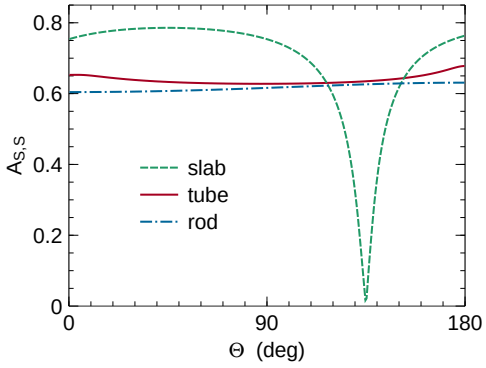

FIG. 3: Absorption factor $A_{S,S}$ for a slab (thickness $d^S = 0.2$, mounted under $\alpha = 135°$), a tube ($R_o^S = 30$, thickness $d^S = 0.1$), and a rod ($R_o^S = d^S = 0.3$); implicit length unit set by $\mu^S = 1$.

outer container; both container parts are hollow cylinders, with thickness d^{Ci} and d^{Cf}, respectively. The two-dimensional integration (5) is carried out numerically, with a hard-coded relative error limit of 10^{-4}.

The *geometry parameters* in the API specification of `absco_tube` and `transmission_tube` are

`double radius,`	R_o^S
`double thickness_S,`	d^S
`double thickness_C_inner,`	d^{Ci}
`double thickness_C_outer`	d^{Co}

Because of the cylindrical symmetry, the transmission does not depend on χ. Therefore, `outgoing_angle` is omitted from the arguments of `transmission_tube`.

A typical curve $A_{S,S}(\Theta)$, without container scattering and without backscattering, is shown in Fig. 3.

D. Program print-absco

In the directory `run/`, a freestanding program `print-absco` is provided that creates a table of absorption coefficients for a given set of common arguments and a given geometry specification. When called as `print-absco -h`, all options and arguments are explained.

V. COMPUTATIONAL DETAILS

This section explains the computations carried out in `libabsco` at a level of detail that is interesting only for those who want to verify or extend the code.

All functions are implemented in double precision. The cut-off constant $\epsilon = 10^{-14}$ is chosen well above the machine roundoff. To keep computation times short, a much lesser precision is required in the numeric integrations.

A. Flat samples

To avoid notational complications, α and Θ are both confined to the interval $[0, \pi]$. A coordinate t is chosen along the normal of the slab. Averages (6) are then computed as

$$\langle f(t) \rangle = d^{-1} \int_0^d \mathrm{d}t\, f(t). \tag{9}$$

For a scattering location in X, given by t, the outgoing path within X has the length

$$p_\mathrm{f}^X(t,\Theta) = \begin{cases} (d-t)/\sin(\alpha-\Theta) & \text{for } \Theta < \alpha, \\ t/\sin(\Theta-\alpha) & \text{for } \Theta > \alpha. \end{cases} \tag{10}$$

The two cases correspond to *transmission* and *reflection*. For $\Theta \simeq \alpha$, the path of the scattered ray within the slab is so long that no meaningful scattering signal is expected; in this case, zeroes are returned as absorption factors. The incoming path is

$$p_\mathrm{i}^X(t) = p_\mathrm{f}^X(t,\pi) = t/\sin\alpha. \tag{11}$$

With the abbreviations $\kappa_\mathrm{i} := \mu_\mathrm{i} d/\sin\alpha$ and $\kappa_\mathrm{f} := \mu_\mathrm{f} d/\sin|\Theta-\alpha|$, the absorption factor for *one* layer is

$$A(\Theta) = \begin{cases} \dfrac{\mathrm{e}^{-\kappa_\mathrm{f}} - \mathrm{e}^{-\kappa_\mathrm{i}}}{\kappa_\mathrm{i} - \kappa_\mathrm{f}} & \text{for } \Theta < \alpha, \\ \dfrac{1 - \mathrm{e}^{-\kappa_\mathrm{i}-\kappa_\mathrm{f}}}{\kappa_\mathrm{i} + \kappa_\mathrm{f}} & \text{for } \Theta > \alpha. \end{cases} \tag{12}$$

In the transmission case, there is a removable singularity at $\kappa_\mathrm{i} = \kappa_\mathrm{f}$ that has no special physical meaning; for $|\kappa_\mathrm{i} - \kappa_\mathrm{f}| < \sqrt{\epsilon}$, the above expression is replaced by the Taylor expansion

$$A(\Theta) \doteq \mathrm{e}^{-\kappa_\mathrm{i}} \left(1 - \frac{\kappa_\mathrm{f} - \kappa_\mathrm{i}}{2} + \frac{(\kappa_\mathrm{f} - \kappa_\mathrm{i})^2}{12}\right). \tag{13}$$

In the composite absorption factors, layers that are fully traversed are accounted for by simple exponential attenuation factors. Similarly, such factors describe the transmission and the attenuation of backscattered neutrons that retraverse the sample.

B. Tubular samples

An algorithm for cylindrical samples has been described in much detail by Paalman and Pings [2], but I found it easier to start from scratch. Everything is based on the computation of $p_\mathrm{f}^Z(\boldsymbol{r}, \Theta)$. Let the scattering plane be spanned by the incoming beam direction $\hat{\boldsymbol{x}}$ and by $\hat{\boldsymbol{y}}$. The incoming beam is scattered in $\boldsymbol{r}$ from where it travels in direction $\hat{\boldsymbol{n}} = \cos\Theta\hat{\boldsymbol{x}} + \sin\Theta\hat{\boldsymbol{y}}$ according to the ray equation $\boldsymbol{s}(\lambda) = \boldsymbol{r} + \lambda\hat{\boldsymbol{n}}$ with $\lambda \geq 0$. Intersections with a cylindrical hull of radius R are given by

$$\lambda = -\boldsymbol{r}\hat{\boldsymbol{n}} \pm \sqrt{R^2 - r^2 + (\boldsymbol{r}\hat{\boldsymbol{n}})^2} \tag{14}$$

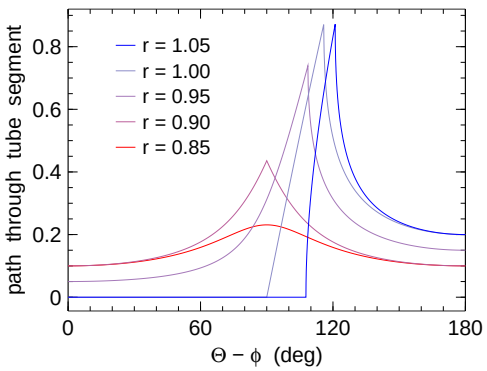

FIG. 4: Path $p_{\rm f}^Z(r, \Theta-\phi)$ in direction Θ through tube segment Z. The starting point $\boldsymbol{r}$ is given by polar coordinates (r, ϕ). Tube dimensions $R_o = 1.0$, $R_i = 0.9$ are fixed; the curves correspond to different values of r.

where we can spell out

$$r^2 - (\boldsymbol{r}\hat{\boldsymbol{n}})^2 = (r_x \sin\Theta - r_y \cos\Theta)^2. \qquad (15)$$

Expressions simplify further if $\boldsymbol{r}$ is given in polar coordinates (r, ϕ): then ϕ and Θ enter results only via their difference; Eq. (14) takes the form

$$\lambda = -r\cos(\Theta - \phi) \pm \sqrt{R^2 - r^2 \sin^2(\Theta - \phi)}. \qquad (16)$$

For an outer hull ($R_o \geq r$), it is easily seen that there is exactly one value $\lambda \geq 0$ (namely the one with sign $+$). For an inner hull ($R_i \leq r$), the expression under the square root in (14) may be negative, in which case there is no intersection. If the square root exists, then it depends on the sign of $\boldsymbol{r}\hat{\boldsymbol{n}}$ whether there are zero or two solutions $\lambda \geq 0$; the special case of two coinciding solutions needs no special consideration.

The geometrical interpretation of all this is simple: if there is no intersection with the inner hull, then the ray leaves the tube directly through its outer hull. If there is a pair of intersections with the inner hull, then the ray leaves the tube towards its inner void, and reenters, before it leaves through the outer hull. The path length $p_{\rm f}^Z(\boldsymbol{r}, \Theta)$ is in both cases easily computed from the respective values of λ. Resulting $p_{\rm f}^Z(r, \Theta - \phi)$ are shown in Fig. 4 for different values of r.

Fig. 5 shows the local absorption factor $f(\boldsymbol{r}, \Theta) = \exp(-\mu(p_{\rm i}(\boldsymbol{r}) + p_{\rm f}(\boldsymbol{r}, \Theta)))$ for a containerless thin tubular sample as function of the polar coordinates (r, ϕ) of the scattering location $\boldsymbol{r}$. In accord with geometric anschauung, f has a rather complex dependence on ϕ, and a relatively smooth dependence on r. Therefore, to compute the average $A(\Theta) = \langle f(\boldsymbol{r}, \Theta)\rangle$ it is preferable to choose r as the inner and ϕ as the outer integration variable.

Both one-dimensional integrations are carried out by the routine `gsl_integration_qags` of the GNU scientific library. "This function applies the Gauss-Kronrod 21-point integration rule adaptively until an estimate of the integral ... is achieved within the desired absolute and relative error limits ... The results are extrapolated using the [Wynn] epsilon-algorithm" [4]. For the inner integration, a hard-coded absolute and relative error limit of $3 \cdot 10^{-5}$ has been chosen; for the outer integration, $1 \cdot 10^{-4}$.

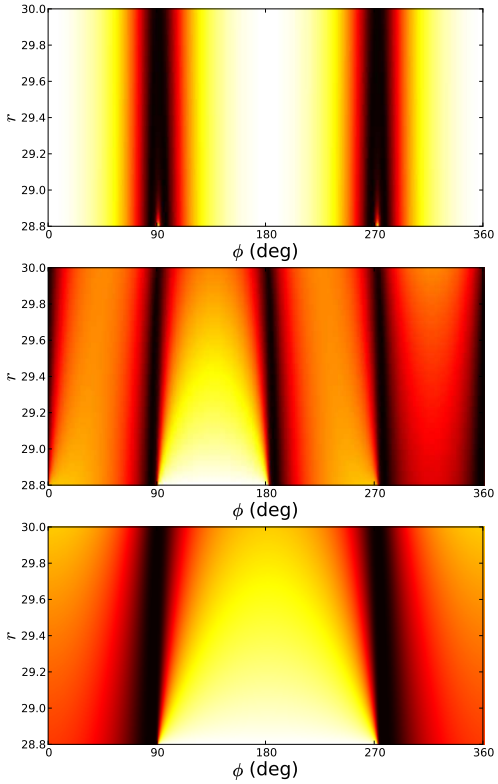

FIG. 5: Local absorption factor $f(\boldsymbol{r}, \Theta) = \exp(-\mu(p_{\rm i}(\boldsymbol{r}) + p_{\rm f}(\boldsymbol{r}, \Theta)))$ for a containerless thin tubular sample with outer radius $R_o^S = 30$, thickness $d^S = 0.2$, attenuation coefficient $\mu^S = 1$, as function of the polar coordinates (r, ϕ) of the scattering location $\boldsymbol{r}$, for three different scattering angles Θ (from top to bottom: 0°, 90°, 180°). Same color scale for all three images: white is 1 (no absorption), black is 0 (complete absorption).

[1] H. L. Ritter, R. L. Harris and R. E. Wood, J. Appl. Phys. **22**, 169 (1951).

[2] H. H. Paalman and C. J. Pings, J. Appl. Phys. **33**, 2635 (1962).

[3] J. Wuttke, Physica B **292**, 194 (2000).
[4] GNU Scientific Library, algorithm QAGS, `http://www.gnu.org/software/gsl/manual/html_node/QAGS-adaptive-integration-with-singularities.html`. This is a reimplementation of QUADPACK code attributed to Piessens, de Doncker-Kapenga, Ueberhuber and Kahaner.