DEMUX/MUX: removal of multiple scattering from small-angle data. By MICHAEL MONKENBUSCH, Institut für Festkörperforschung des Forschungszentrums Jülich, D-5170 Jülich, Germany

(Received 1 October 1990; accepted 30 April 1991)

Abstract

With DEMUX and MUX, a pair of subroutines is provided that allows for the straightforward removal of multiple scattering from small-angle scattering data (DEMUX) or the simulation of the effect of multiple scattering (MUX). The data may be given in an arbitrary scaling. The parameter that must be given to fix the amount of multiple scattering is the transmission reduction of the sample due to small-angle scattering. The algorithm used is based on the method of Schelten & Schmatz [J. Appl. Cryst. (1980), 13, 385–390]. It is valid for a plane flat sample exhibiting scattering of significant intensity at small angles only.

Introduction

Unlike the general multiple-scattering problem, which is only accessible via Monte Carlo simulation of the scattering process, the predominantly elastic small-angle scattering from a plane sample oriented normal to the beam may be solved in closed form using Fourier integrals. The simplicity of this case depends on the assumptions $\cos \theta = 1$ and $\sin \theta = \sin(\theta_1 + \theta_2) = \sin \theta_1 + \sin \theta_2$, with $\theta = \text{final scattering angle}$. The first assumption ensures that all radiation particles have a path length through the sample which is approximately equal to the sample thickness, $d$, independent of the number of small-angle-scattering events. Therefore, extinction along all paths may be accounted for by a common factor $\exp{-\mu d}$. $\mu$ is the total small-angle-scattering cross section per unit sample volume. The second assumption yields the result that multiple-scattering processes may be described as repeated convolutions of the single-scattering probability on the two-dimensional space spanned by two orthogonal scattering angles ($\theta_x, \theta_y \sim q$). The use of the convolution theorem while adding the multiple-scattering contributions in the Fourier-transformed scattering-plane space results in the conclusion that the two-dimensional Fourier transform of the scattering intensity with multiple scattering equals the exponential function of the Fourier-transformed scattering probability (Schelten & Schmatz, 1980). To get rid of the necessity to have an absolute gauge of the scattering data, the algorithm is formulated in such a way that only the transmission, $T$, due to small-angle scattering enters the procedure as an absolute number. It is assumed that other attenuation processes due to true absorption and/or incoherent and/or large-angle scattering have already been accounted for. With $h(q)$ as scattering intensity including multiple scattering and $s(q)$ as scattering intensity due to single scattering only and without transmission attenuation and $h(r)$ and $s(r)$ the 2D Fourier transforms, the analog to equation (14) of Schelten & Schmatz reads

$$h(q) = -[s(r = 0)/\ln T] \times F^{-1}\left(\exp\left[-\ln T[s(r = 0)] + \ln T\right] - 1\right).$$

(1)

$F^{-1}$ is the inverse 2D Fourier transform. It is obvious that the above equation may be solved to yield the single scattering intensity, $s(q)$:

$$s(q) = [h(r = 0)/(1 - T)] \times F[\ln\left[h(r)/h(r = 0)\right](1 - T) + T].$$

(2)

The use of these equations allows for the treatment of scattering data ($s$ or $h$) with arbitrary scaling; they produce the scattering of the same sample in the same apparatus with the same scaling but with or without the effect of multiple scattering (including attenuation). Any absolute gauging may be done before or after the multiple-scattering correction. The required information on the total scattering probability is completely contained in $T$.

It should be noted that the removal of multiple scattering using (2) is not as badly behaved as a usual deconvolution. The multiple scattering is usually represented by a smoothly varying intensity of at most 10% of the total scattering intensity. The above equation extracts this multiple scattering from the data and implicitly subtracts it. Any error in the determined multiple-scattering intensity will be an error of a value which contributes only about 10% to the scattering. However, in special cases where the single scattering is nearly or equal to zero but the multiply-scattered intensity is large, small errors may lead to large relative errors in the corrected data. Nevertheless, the example given below exhibits a very well behaved operation of DEMUX.
For an isotropic scatterer the 2D Fourier transforms may be written as

\[ h(r) = F[h(q)] = \int_0^\infty j_0(qr) dq. \]  

(3)

Data from an area detector may be treated by direct use of 2D Fourier transforms. The discrete FFT (fast Fourier transform) algorithms implemented in many mathematical subroutine libraries are a very efficient tool to perform these Fourier transforms. In fact, even the isotropic problem may be solved more efficiently and accurately by use of the 2D FFT on the scattering intensities distributed on a 2D data field representing the cells of a hypothetical area detector (see Errors).

For the removal of multiple scattering from real scattering, data intensity values have to be known down to \( q = 0 \), therefore usually an extrapolation to fill the data gap due to the central beam stop has to be employed before using (2). For this extrapolation a parabolic shape of the scattering curve symmetric around \( q = 0 \) may be assumed. Due to the 2D nature of the transforms the accuracy of the extrapolation is not very crucial because the affected area is small compared to the complete data area.

At the high-\( q \) side the data are extrapolated as \( aq^{-2} \) if data values beyond the measured range are addressed by the algorithm.

The program

The program consists of two Fortran subroutines, \textit{MUX} and \textit{DEMUX}, which implement the above formulas for a one-dimensional scan from an isotropic scatterer. Because they employ a 2D FFT routine for the transforms, it is trivial to create entries for true 2D area-detector data. Because scattering data are all symmetric with respect to the origin, it is sufficient to perform real cosine transforms. The present implementation also contains the subroutine \textit{RFFT2D}, which extends the linear cos FFT, \textit{FCOST}, from the IMSL Library (IMSL, 1987) to a 2D transform. \textit{FCOST} performs the following summation:

\[ h_n = \frac{1}{2} \sum_{m=1}^{N-1} s_m \cos[(m-1)(n-1)\pi/(N-1)] + s_n + s_n(-1)^{m-1}. \]  

(4)

Therefore, \textit{RFFT2D} takes one quadrant of the \( q \) data area with the \( q = 0 \) intensity as \((1,1)\) element as input. All elements (inner and boundary) are taken with unit weight to give the real part of the Fourier transform of the whole area. \textit{RFFT2D} is its own inverse. \textit{RFFT2D} or \textit{FCOST} may be replaced by any corresponding FFT routine from other available libraries.

The calls for both routines \textit{MUX} and \textit{DEMUX} are the same:

\begin{verbatim}
call DEMUX(Xin,Yin,Nin,T,Xmax,Nfft,Jout,Yout,Yout,Jer).
\end{verbatim}

\( Xin \) and \( Yin \) are input vectors containing \( Nin \) points of scattering data in ascending \( X = q \) order, leading and trailing zero data points are removed within \textit{DEMUX}. The \( q \rightarrow 0 \) extrapolation uses the first three nonzero data points to determine the slope and value of the extrapolation parabola at the lowest measured angle. The high-\( q \) extrapolation uses the last three data points for that purpose. \( T \) is the transmission due to small-angle scattering. An adequate method to determine \( T \) would be to create a data set containing scattered data including the transmitted primary beam on the same scale. Then the primary-beam intensity may be determined by subtracting the extrapolated scattered intensity. The area integral of this primary intensity divided by the area integral of the complete data set is the required transmission, provided the \( q \) range was large enough to include all significant small-angle scattering. \( \text{Xmax} (= q_{\text{max}}) \) is the range up to which data are included in the calculation; it equals the range of output data. However, the valid output data range may be somewhat less. \( \text{Nfft} \) is the number of points used in one dimension of the 2D FFT summations. For neutron small-angle data \( \text{Nfft} = 128 \) is sufficient in most cases. Due to performance reasons \( \text{Nfft} \) should be a power of 2. \( \text{Jer} \) controls the amount of written output from the subroutine. \( Xin, Yin, Jout, Yout \) contain the result. They may share the same storage locations with \( Xin \) and \( Yin \). The number of valid data points is \( \text{Nfft} \). \( \text{Jer} = 0 \) indicates execution without error.

\section*{Errors}

\textbf{Erroneous transmission values}

A wrong value for the transmission due to small-angle scattering, \( T \), leads to a systematic error in the corrected data. The relative amount of single scattered intensity to the total scattering intensity is

\[ y = T \ln T/(T - 1). \]  

(5)

With \( \hat{h}_1 \) as normalized single-scattering intensity and \( \hat{h}_2 \) as normalized multiple-scattering contributions the total scattering is

\[ \hat{h}_{\text{tot}} = y\hat{h}_1 + (1 - y)\hat{h}_2 \].  

(6)

The normalization for \( \hat{h}_1 \) is chosen such that the integral scattering equals the total scattering integral. A small error in the value of \( T \) will not change the form of \( \hat{h}_1 \) and \( \hat{h}_2 \) very much but will influence their relative weights. The error in the single-scattering intensity as computed by \textit{DEMUX} will therefore be

\[ \Delta h_1 = [(h_1 - h_{1-\text{err}})\Delta y/y = h_1(1 - y)\Delta y/y] \]  

(7)

where \( h_1 \) are the actual contributions of the different scattering processes to the original data.

\section*{Statistical errors of the input data}

Unlike deconvolution procedures the removal of multiple scattering does not suffer from noise amplification. The procedure virtually removes the multiple-scattering contribution determined by some kind of implicit integration from each data point and multiplies the data by \( 1/T \). Because the data to be removed are integrated values their statistical error should be low, therefore most of the noise in the corrected data is the noise of the original data multiplied by \( 1/T \). Nevertheless, the relative noise may be much larger in the corrected data than in the original data if a region is considered where
most of the original intensity was due to multiple scattering.

**FFT sampling and aliasing errors**

The use of discrete Fourier sums as computed by the FFT algorithms introduces some errors compared to Fourier integrals. However, the differences are well known and their effect may be estimated. The sampling of equidistant points may be considered as the effect of multiplication of a continuous function with an equidistant sequence of delta functions. The convolution theorem yields the result that is the convolution of the integral transform with another delta function. Only if the integral transform is zero for 'frequencies' larger than half of the distance of these delta functions do series and integral coincide. Otherwise the integral transform functions associated with different delta functions do overlap and the series value in the overlap region is the sum of contributions of different 'copies' of the integral transform. If the data are 'smooth' the transform will drop very quickly and the overlapping contributions are small. But even with finite contributions in this region the method may still work quite accurately because even with large overlap the FFT and its inverse applied to any data yield an exact copy of the input data. The nonlinear operation on the transformed data may disturb this compensation. However, if the transform is small in the overlap region, the nonlinear (exp or log) functions are taken for values proportional to \(1 - s/s(0)\) where \(s\) is the transform. For small \(s/s(0)\) the functions may be linearized, then the overlap region is additive and the balancing done by the inverse transform still works. As a consequence the method does not require \(s = 0\) in the overlap region but only \(s/s(0) \ll 1\) for the method to work properly. This is the reason why the FFT method performs better than direct one-dimensional numerical integration using Bessel functions.

For proper operation of \(MUX\) it should be observed that the data \(q\) range will spread to some extent by multiple-scattering events if original data on a finite interval are considered. The \(MUX\) range, \(X_{\text{max}}\), has to be chosen such that it is large enough to cover the \(q\) range after multiple scattering. Otherwise, the intensities for \(q > X_{\text{max}}\) will be added to the data in the interval \([0,X_{\text{max}}]\) by the above-described overlap in the form of a mirror image of the data above \(X_{\text{max}}\) at the interval boundary.

The use of \(DEMUX\) will not spread the data but requires original data taken up to \(X_{\text{max}} = q_{\text{max}}\), where the intensity has dropped to negligible magnitude. The original data interval has to contain the region where significant multiple-scattering contributions to the data in the region of interest may stem from.

**System requirements**

The \(MUX\) and \(DEMUX\) subroutines have to be called by a main program that has to be provided by the user. This main program has to supply the scattering data in two linear arrays, \(X_{\text{in}}\) and \(Y_{\text{in}}\). After execution, treated data are returned to \(X_{\text{out}}\) and \(Y_{\text{out}}\), which may be the same as \(X_{\text{in}}\) and \(Y_{\text{in}}\). The subroutines (excluding the IMSL routine) require 14968 bytes of code and have an array space usage of \(M_{\text{dim}} \times (M_{\text{dim}} + 1)\) and the IMSL routine \(\text{FCOST}\) needs an additional workspace of \(3 \times M_{\text{dim}} + 15\) REALS. \(M_{\text{dim}}\) is the maximum value of \(N_{\text{fft}}\); it may be simply changed by changing the corresponding \(\text{PARAMETER}\) statements in the \(MUX\) and \(DEMUX\) subroutines. The implementation with \(M_{\text{dim}} = 512\) therefore needs 264 207 REALS = 1 056 828 bytes array space. However, useful work may be done even with a version with \(M_{\text{dim}} = 128\), which needs only 67 644 bytes array space.

The code is written in Fortran77 using the DO ... ENDDO construction as extension. The time for one call
of DEMUX with \(N_{\text{fft}} = 128\) takes 3.14 s CPU time on an IBM3081D32 with the VM/CMS operating system. A call with \(N_{\text{fft}} = 512\) takes 19.2 s. For a standard IBM PC-AT with coprocessor these times should be multiplied by a factor of about 150.

The program is available on request from the author.

**Example**

The use of DEMUX is illustrated by the treatment of neutron small-angle data from an AOT microemulsion \([\text{AOT} = \text{sodium 1,4-bis(2-ethylhexyl) sulfosuccinate}]\). The data have been measured at the KWS1 neutron small-angle scattering machine by S. Komura, who kindly gave them as test data. Data from samples with 1 and 2 mm thickness corresponding to transmissions of 0.65 and 0.42 have been evaluated. The treated data were fitted to

\[
l(q) = (a_0 + a_2q^2 + a_4q^4)^{-1}
\]

which has been shown to be valid for many surfactant systems by Teubner & Strey (1987). After treatment and normalization with respect to thickness the data should be the same for both sample thicknesses. Fig. 1 shows the normalized original data already extrapolated to \(q \to 0\) compared with the DEMUX results for both thicknesses. As anticipated, both treated curves coincide very well, whereas the original data differ due to the different amount of multiple scattering. The regions to the right and left of the peak are obviously much affected by multiple scattering. The residual difference of both corrected curves probably stems from inaccuracies of the transmission figures and of the \(q \to 0\) extrapolation. The two results are equal within the size of one symbol illustrating the observation that \(N_{\text{fft}} = 128\) is sufficient for most cases.

Provided the theory of Teubner & Strey holds the corrected data should be describable by (8). Fig. 2 shows the result of such a fit to both (1 and 2 mm) corrected data. Compared to a fit to the 2 mm original data, shown in Fig. 3, a much better agreement could be obtained after the operation of DEMUX. The fit parameters \(a_0, a_2\) and \(a_4\) differ by about a factor of two.

I thank Professor S. Komura for his kindness in giving his scattering data as an example of the use of the DEMUX program.

**References**


**RIBBONS 2.0.** By MIKE CARSON, University of Alabama at Birmingham, Center for Macromolecular Crystallography, 252 BHS, 79 THT University Station, Birmingham, AL 35294, USA

(Received 5 December 1990; accepted 18 June 1991)

**Abstract**

The program RIBBONS 2.0 allows real-time viewing of solid shaded ribbon models of macromolecules. The primary features of the software are the ability to create a wide variety of styles of ribbon drawings interactively and to toggle between various coloring schemes chosen to reflect assorted geometrical and biochemical properties. Spheres, cylinders, dots, polygons and text are also supported. The auxiliary programs included make RIBBONS 2.0 a powerful tool for visual structural analysis as well as for presentation graphics. The program is currently available only for the Silicon Graphics 4D series of workstations. A port to the Evans & Sutherland ESV workstation employing PEX is under development.

© 1991 International Union of Crystallography