Basic numerical and statistical methods

Vladimir Volkov
Institute of Crystallography, Moscow, Russia

Plan of the talk:
1. Introduction: data interpretation
2. Statistics: errors and goodness-of-fit
3. Data fitting by simple functions, solution of system of linear equations.
4. Data fitting by nonlinear models, methods for minimization.
5. Fourier transform, basic ideas.
Main goal for end data interpretation

to build a mathematical description of a physical model,

calculate physical responses $Y(x)$ from the model that could be measured in an experiment and

compare them with experimental data (what is goodness-of-fit?)

Then, correct the model parameters by using the comparison results (methods for automatic correction?)

Case study: structure refinement by molecular tectonics.
Case study: PDC in the crystal and in solution

Left: crystallographic model (interaction area, 13.44 nm²)
Right: result of rigid body refinement (r.m.s., 0.58 nm)

*J. Biol. Chem.* 275, 297-303.
**Goodness-of-fit**

**Poor fit:**
Differences between experimental data and the model are large and systematic. The goal is to find model parameters that provide minimum difference.

**Good fit:**
Differences are less than above. In addition, they do not have systematic behaviour. This means, that the model *may be* adequate.

Merit function: analysis of residuals.
If sum over all occurrences = 1 (normalization) then this plot relates to the probability density function: $P\{Y=y\}$.
Number of impulses counted by a detector: Poisson distribution.

Poisson random numbers with mean 50

Middle box value

Asymmetric at low mean values

Almost symmetric, close to a normal distribution at large mean values
Statistical characterization of a data set

Most probable (expected) value of a random variable $y_i$:

Mean value: $\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$
from $\sum_{i=1}^{N} y_i = \sum_{i=1}^{N} \bar{y} = N \bar{y}$

Mean value from density distribution histogram:
$\bar{y} = \frac{\sum_{j} b_j n_{b_j}}{\sum_{j} n_{b_j}}$

Measure of variance

Mean linear deviation:
$d = \frac{1}{N} \sum_{j=1}^{N} |y_j - \bar{y}|$

Variance (empirical):
$\sigma^2 = \frac{1}{N-1} \sum_{j=1}^{N} (y_j - \bar{y})^2$

Standard deviation: $\sigma$

(measure of variability about mean value: $\bar{y} \pm \sigma$)
**Poisson random variable** (discrete), which is the number of successes that will be observed in repeated intervals of time, each of which may result in the occurrence or nonoccurrence of a success.

\[ P\{Y = y\} = \frac{Lt^y}{y!} e^{-t} \]  

(probability that randomly obtained \( Y \) equals to the prescribed value \( y \))

where \( y \) is the fixed value, \( t \) is the observation time interval, \( Lt \) is the mean number of successes per interval of measurement \( t \).

mean value (expected value) \( E\{ Y \} = Lt \), variance = \( Lt \) (thus, we measure values \( Y = Lt \pm \sqrt{Lt} \ )

The larger is \( t \), the less is the relative error.

**Normal random variable** (continuous).

De Moivre-Laplace's theorem:

\[
\lim_{N \to \infty} \left[ c < \frac{Y - Lt}{\sqrt{Lt}} < d \right] = \frac{1}{\sqrt{2\pi}} \int_d^c e^{-x^2/2} \, dx
\]

Most of statistical estimates uses this fact.

Common formula:

\[
N(a, \sigma^2) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\cdot\sigma^2}}
\]

mean value (expected value) \( E\{ Y \} = a \), variance = \( \sigma^2 \).
Measure of degree of adjacency of two data sets \(a_1, a_2, \ldots, a_N\) and \(b_1, b_2, \ldots, b_N\)

Our goal: to get a single characteristic number.

We can not use summation over all deviations because they may have opposite signs and compensate each other giving a result near zero.

But we may use the sum over absolute values of deviations:

\[
d = \frac{1}{N} \sum_{j=1}^{N} |a_j - b_j|
\]

It works, but the use of this formula leads to significant mathematical difficulties: this function is not differentiable.

Squaring the deviations proves to be a much more attractive solution:

\[
d^2 = \frac{1}{N - 1} \sum_{j=1}^{N} (a_j - b_j)^2
\]
Weighting of the summation terms: equalization of influence of errors

An example: level of Poisson noise is proportional to square root of the intensity.

To reduce the influence of data points with large associated errors, all the elements are divided by their corresponding deviates. These deviates must be determined independently.

\[
\frac{1}{N-1} \sum_{j=1}^{N} \left( \frac{A_j - B_j}{\sigma_j} \right)^2
\]

A: Curve to be fitted to

B': curve minimizing

B'': curve minimizing

E: Errors in A

relatively small       large
Chi-square: least squares fitting of a set of $N$ data points $A_j$ with errors $\sigma_j$.

$$\min \left[ \chi^2 = \frac{1}{N - M - 1} \sum_{j=1}^{N} \left( \frac{A_j - B_j}{\sigma_j} \right)^2 \right]$$

Calculated from a model described by a set of $M$ parameters $p_k$.

We use here the term “chi-square”, because sum of squares of random deviates have chi square distribution. Expected minimum value of chi-square is about unity.

Main idea of a fitting procedure:

At the point of the minimum, the derivatives (gradients)

$$\frac{\partial \chi^2}{\partial p_k} = 0$$

because we have here zero slope of the surface of the minimizing function.
We consider the problem of fitting a set of \( N \) data points \((x_i; y_i)\) to a straight-line model with parameters \( a \) and \( b \):

\[
y(x) = y(x; a; b) = a + bx
\]

This problem is often called **linear regression**. We assume that the uncertainty \( \sigma_i \) associated with each measurement \( y_i \) is known, and that the \( x_i \)'s (values of the dependent variable) are known exactly.

To measure how well the model agrees with the data, we use the chi-square merit function, which in this case is

\[
\chi^2 = \frac{1}{N-1} \sum_{j=1}^{N} \left( \frac{y_j - a - bx_j}{\sigma_j} \right)^2
\]

At its minimum, derivatives with respect to \( a; b \) vanish.

\[
0 = \frac{\partial \chi^2}{\partial a} = -2 \sum_{j=1}^{N} \frac{y_i - a - bx_j}{\sigma_j^2}
\]

\[
0 = \frac{\partial \chi^2}{\partial b} = -2 \sum_{j=1}^{N} \frac{x_j (y_i - a - bx_j)}{\sigma_j^2}
\]
Fitting data by a straight line.

These conditions can be rewritten in a convenient form if we define the following sums:

\[ S = \sum \frac{1}{\sigma_j^2}, \quad S_x = \sum \frac{x_j}{\sigma_j^2}, \quad S_y = \sum \frac{y_j}{\sigma_j^2}, \quad S_{xx} = \sum \frac{x_j^2}{\sigma_j^2}, \quad S_{xy} = \sum \frac{x_j y_j}{\sigma_j^2}, \quad \Delta = SS_{xx} - (S_x)^2 \]

With these definitions, we have here the system of linear equations:

\[ \begin{align*}
    aS + bS_x &= S_y \\
    aS_x + bS_{xx} &= S_{xy}
\end{align*} \]

This system may be solved in any common way giving

\[ a = \frac{S_{xx}S_y - S_xS_{xy}}{\Delta}, \quad b = \frac{S_xS_{xy} - S_{xx}S_y}{\Delta} \]

Now, we must estimate the probable uncertainties in the estimates of \( a \) and \( b \), since obviously the measurement errors in the data must introduce some uncertainty in the determination of those parameters. If the data are independent, then each contributes its own bit of uncertainty to the parameters.
Fitting data by a straight line.
Propag. of errors in solution.

Consideration of propagation of errors shows that the variance in the value of any function will be

\[ \sigma_i^2 = \sum_{j=1}^{N} \sigma_j^2 \left( \frac{df}{dy_j} \right)^2 \]

\(i\): index of a model parameter,

\(j\): index of an experimental point

For the straight line,

\[ \sigma_a^2 = \frac{S_{xx}}{\Delta} \]

\[ \sigma_b^2 = \frac{S}{\Delta} \]

Thus, we have the solution for coefficients of a straight line:

\[ a \pm \sigma_a, \quad b \pm \sigma_b \]

Because we minimize sum of squares, it is possible to obtain once more important estimate - the goodness-of-fit of the data to the model. This estimate indicates whether the parameters \(a\) and \(b\) in the model have any meaning at all! The probability \(Q\) that a value of chi-square as poor as the value we use to derive the solution,

\[ \chi^2 = \frac{1}{N-1} \sum_{j=1}^{N} \left( \frac{y_j - a - bx_j}{\sigma_j} \right)^2 \]

should occur by chance is

\[ Q = \Gamma^{inc} \left( \frac{N-2}{2}, \frac{\chi^2}{2} \right) \]

where \(Q = \Gamma^{inc}(p, q)\) is incomplete gamma function. Using known tables of statistics one may find that if \(Q > 0.1\) then the solution is reliable, if \(0.001 < Q < 0.1\) then \(a\) and \(b\) may be acceptable. If \(Q < 0.001\) then the model and/or estimation procedure can rightly be called into question.
Fitting data by polynomials

Polynom: \( F(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m \)

What we minimize: \( \chi^2 = \frac{1}{N - m - 1} \sum_{j=1}^{N} \left( \frac{y_j - F(x_j)}{\sigma_j} \right)^2 \)

Condition of minimum: \( \frac{\partial \chi^2}{\partial a_j} = 0, \quad j = 0, \ldots, m \)

System of \( m \) linear equations:

\[
\begin{align*}
    a_0 \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j &+ a_1 \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j + \ldots + a_m \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j^m = \sum_{j=1}^{N} \frac{1}{\sigma_j^2} y_j \\
    \ldots & \quad \ldots \\
    a_0 \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j &+ a_1 \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j + \ldots + a_m \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j^{m+1} = \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j y_j \\
    a_m \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j^m &+ a_1 \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j^{m+1} + \ldots + a_m \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j^{2m} = \sum_{j=1}^{N} \frac{1}{\sigma_j^2} x_j^m y_j
\end{align*}
\]
Systems of linear equations - matrix notation

Matrix notation: $C A = B,$
solution: $A = C^{-1}B$

Errors in solution: $\sigma^2 a_j = \chi^2 C^{-1}_{jj}$
Solution of systems of linear equations

Method of Gauss-Jordan elimination (triangulation)

Direct move:

1) \( C_{11}a_1 + C_{12}a_2 + C_{13}a_3 = b_1 \)
2) \( C_{21}a_1 + C_{22}a_2 + C_{23}a_3 = b_2 \)
3) \( C_{31}a_1 + C_{32}a_2 + C_{33}a_3 = b_3 \)

multiply 1) by \( C_{21}/C_{11} \), subtract from 2);
multiply 1) by \( C_{31}/C_{11} \), subtract from 3);

1) \( C_{11}a_1 + C_{12}a_2 + C_{13}a_3 = b_1 \)
2) \( C^*_2 a_2 + C^*_3 a_3 = b^*_2 \)
3) \( C^*_3 a_3 = b^*_3 \)

multiply 2) by \( C^*_3/C^*_2 \), subtract from 3);

1) \( C_{11}a_1 + C_{12}a_2 + C_{13}a_3 = b_1 \)
2) \( C^*_2 a_2 + C^*_3 a_3 = b^*_2 \)
3) \( C^*_3 a_3 = b^*_3 \)

Backward substitution:

\( a_3 = b^*_3 / C^*_3 \) from 3);
\( a_2 = (b^*_2 - C^*_3 a_3) / C^*_2 \) from 2);
\( a_1 = (b_1 - C_{13} a_3 - C_{12} a_2) / C_{11} \) from 1)

“Pivoting”:

Avoiding deletion by small coefficients by rearrangement of rows.
Solution of systems of linear equations

Overestimated systems: least squares solution.

This system does not have an exact solution.

Case study: fitting data by polynomials
(N equations):

\[ a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_m x_i^m = y_i \]

If we look at the matrix \( C \), we find that

\[ C = D^TD \quad \text{(matrix of normal equations)} \]

Solution by least squares:

\[
C A = D^T B, \\
A = (C)^{-1} D^T B
\]
Solution of systems of linear equations

Singular value decomposition (SVD)

\[ D = U \Sigma V^T \]

Orthonormal matrices

\[ U^{-1} = U^T; \quad V^{-1} = V^T \]

Identity matrix (E)
Solution of systems of linear equations

Solution by least squares using normal equations:
\[ D^T D \, A = D^T \, B, \]
\[ A = (D^T D)^{-1} \, D^T \, B \]

Solution by least squares using SVD:
\[ D = U \, S \, V^T; \]
\[ U^T \, U \, S \, V^T \, A = U^T \, B; \quad \text{because} \quad U^T U = E, \quad \text{and} \quad E \text{ play role of a unity in multiplications:} \]
\[ S \, V^T \, A = U^T \, B; \quad S^{-1} S = E, \text{ then} \]
\[ V^T \, A = S^{-1} \, U^T \, B; \quad \text{critical operation - inversion of} \quad S_{ii} \]
\[ A = V \, S^{-1} \, U^T \, B \quad \text{because} \quad V \, V^T = E \]
\[ A = \sum_{i=1}^{M} \left( \frac{U(i) \cdot B}{S_i} \right) V(i) \pm \frac{1}{S_1} \, V(1) \pm \cdots \pm \frac{1}{S_m} \, V(m) \]

\[ \sigma^2_{a_j} = \sum_{i=1}^{M} \left( \frac{V_{ji}}{S_i} \right)^2 \]
Minimization of functions

You are given a single function \( f(x) \) that depends on one or more independent variables.

You want to find the value of those variables \( \{x\} \) where \( f(x) \) takes on a minimum value. You can then calculate what value of \( f(x) \) is achieved at the minimum.

An extremum (minimum point) can be either \textit{global} (truly the lowest function value) or \textit{local} (the lowest in a finite neighborhood and not on the boundary of that neighborhood).
Minimization of functions

One-dimensional search:
Golden section approach

Successive bracketing of a minimum. The minimum is originally bracketed by points 1, 3, 2. The function is evaluated at 4, which replaces 2; then at 5, which replaces 1; then at 6, which replaces 4. The rule at each stage is to keep a center point that is lower than the two outside points. After the steps shown, the minimum is bracketed by points 5, 3, 6.

Golden proportion:

\[
\frac{b}{a+b} = \frac{a}{b} \approx 0.618...
\]

The main idea: at each step, calculate only one function value with reducing the interval of uncertainty.
Minimization of functions

One-dimensional search:

Golden section with parabolic interpolation (Brent's Method)

It is often much more effectivet than the successive golden section.

At any particular stage, it is keeping track of 5 function points (not necessarily all distinct), \( a, b, u, v, w \) and \( x \), defined as follows:

- The minimum is bracketed between \( a \) and \( b \);
- \( x \) is the point with the very least function value found so far (or the most recent one in case of a tie);
- \( w \) is the point with the second least function value;
- \( v \) is the previous value of \( w \);
- \( u \) is the point at which the function was evaluated most recently.

At the each particular step we first try to find the minimum of a parabola passing through the three best points and use golden section if the minimum point does not reduce the interval better.
Minimization of functions

Multidimensional search.

1. Direct methods (only function values are used): random search, downhill simplex method due to Nelder and Mead

2. Gradient methods: conjugate gradients, variable metrix

3. Nonlinear Least Squares: algorithm NL2SOL.

3. Global minimization
Minimization of functions

Downhill simplex method due to Nelder and Mead
Possible outcomes for a step in the downhill simplex method in a three-dimensional case $f(x_1,x_2,x_3)$: the simplex at the beginning of the step, here a tetrahedron, is shown, top. The simplex at the end of the step can be any one of (a) a reflection away from the high point, (b) a reflection and expansion away from the high point, (c) a contraction along one dimension from the high point, or (d) a contraction along all dimensions towards the low point. An appropriate sequence of such steps will always converge to a minimum of the function.
Minimization of functions
Steepest descent method
(most reasonable but with worst behaviour!):

Gradient vector
Descent vector
Next gradient

Successive minimizations along coordinate directions in a long, narrow “valley” (shown as contour lines). Unless the valley is optimally oriented, this method is extremely inefficient, taking many tiny steps to get to the minimum, crossing and re-crossing the principal axis.
Minimization of functions

Conjugate gradient approach

Taylor series approximation of any function:

\[ f(x) = f(P) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \ldots \approx c - g \cdot x + \frac{1}{2} x \cdot A \cdot x \]

Function value at the current point \( P \)

- \( g \) : antigradient vector

The matrix \( A \) whose components are the second partial derivative matrix of the function is called the Hessian matrix of the function at \( P \).

Conjugate conditions for \( d \) (vectors of search steps)

\[ d_i \cdot A \cdot d_j = 0 \quad \text{with} \quad i \neq j \]

Rule for generating directions

\[
\begin{align*}
\begin{cases}
  d_0 & = -g_0 \\
  d_k & = -g_k + \beta_k d_{k-1}
\end{cases} \\
\begin{cases}
  x_{k+1} & = x_k + \alpha_{k+1} d_k \\
  \alpha_{k+1} & = \min_{\alpha} \{ x_k + \alpha d_k \}
\end{cases}
\end{align*}
\]

Local search along \( d_k \)

Rule for conjugate gradient multiplier

\[
\beta_k = \frac{g_k A d_{k-1}}{d_{k-1} A d_{k-1}} = \frac{g_k (g_k - g_{k-1})}{d_{k-1} (g_k - g_{k-1})}
\]
Minimization of functions

Conjugate gradient approach

Antigradient direction (steepest descent) \(-g_1\)

Conjugate direction
\[ d_1 = -g_1 + \beta_1 d_0 \]
Minimization of functions

Newton method: $A$ is known

$$f(x) = c - g \cdot x + \frac{1}{2} x \cdot A \cdot x$$

$$x_{\min} = x_0 - A^{-1} g_0$$

No linear search is required for quadratic functions.
Minimization of functions

Quasi-Newton methods

\[ f(x) = c - g \cdot x + \frac{1}{2} x \cdot A \cdot x \]

min. \[ x_{\text{min}} = x_0 - A^{-1} g_0 \]

\[ A^{-1} = \sum_{k=1}^{M} \frac{p_k \times p_k}{p_k y_k} \]

\[ d_0 = -g_0 \]

Newton step

\[ d_k = -A_k g_k \]

Accumulation of A

\[ A_k = A_{k-1} + U_{k-1} \]

\[ x_{k+1} = x_k + \alpha_k d_k \]

step to min along \( d_k \)

\[ y_k = g_k - g_{k-1} \]

difference gradient

\[ p_k = x_k - x_{k-1} = \alpha_{k-1} d_{k-1} \]

search path

Common formula for quasi-Newton corrections:

\[ U_k = \frac{y_k \times y_k}{p_k p_k} - \frac{A_{k-1} p_k}{p_k A_{k-1} p_k} + \varphi_k \cdot p_k A_{k-1} p_k \cdot \omega_k \times \omega_k \]
Search paths for different minimization methods

Nelder-Mead
(each 5-th step)
100 iterations/
260 function
evaluations

Steepest descent
(no success)

Conjugate
directions
30/217

Quasi-
Newton
20/140
Algorithm NL2SOL for nonlinear least squares problems

Consider vector function $r(x)$ which values are elements of discrepancy between two data sets: $y_j$ and $F(x_j), j=1,\ldots,M$. Here $x_i$ are the model parameters, $i=1,\ldots,N$.

Chi-square is computed as

$$f(x) = \frac{1}{2} r \cdot r$$

The Jacobian matrix

$$J = \frac{\partial r_i}{\partial x_j}$$

Gradient at the point $x$

$$g = \frac{\partial f(x)}{\partial x} = \frac{\partial}{\partial x} \left[ \frac{1}{2} r \cdot r \right] = \frac{\partial}{\partial x} \left[ \frac{1}{2} \sum_{i=1}^{M} r_i^2 \right] = J^T \cdot r$$

Hessian matrix for $f(x)$

$$\frac{\partial^2 f(x)}{\partial x_j^2} = \sum_{i=1}^{M} \begin{bmatrix} \frac{\partial r_i}{\partial x_j} & \frac{\partial r_i}{\partial x_j} \end{bmatrix}^T + r_i \frac{\partial^2 r_i}{\partial x_j^2} = J^T J + S$$

NL2SOL uses Quasi-Newton scheme where $S$ is accumulated by quasi-Newton formulae and $J$ is computed directly at each step. The approximation to $A$ in this case is much better. This is the most powerful known algorithm for least-squares problems (TENSOLVE is better?).
Main idea of the simulated annealing: let the value of \( f(x) \) is the objective function. The system state is the point \( x \). The control parameter \( T \) is something like a temperature, with an annealing schedule by which it is gradually reduced. And there must be a generator of random changes in the parameter values, a procedure for taking a random step from \( x_i \) to \( x_{i+1} \). Offered a succession of options, a simulating annealing method is assumed to jump to the new point with probability \( P = \exp[(f_i - f_{i+1})/kT] \). Notice that if \( f_i > f_{i+1} \), this probability is greater than unity; in such cases the change is arbitrarily assigned a probability \( P = 1 \), i.e., the system always took such an option. This general scheme, of always taking a downhill step while sometimes taking an uphill step, has come to be known as the Metropolis algorithm. In other words, the system sometimes goes *uphill* as well as downhill; but the lower the temperature, the less likely is any significant uphill excursion. The schedule of the algorithm is in successive decreasing the annealing temperature during the search. Finally, it takes only successful steps.

Simulated annealing tries “random” steps; but in a long, narrow valley, almost all random steps are uphill and this algorithm is able to accept them thus continuing the minimization.
Simulated annealing approach based on the downhill simplex method due to Nelder and Mead

The implementation of the Metropolis simulated annealing is here as follows: we ADD a [logarithmically] distributed random number, proportional to the temperature T, to the STORED vertex function value each time we look at it, and we SUBTRACT a random number from the function value calculated for every NEW point that is tried as a replacement candidate. This addition forces the simplex to randomly fluctuate keeping its size proportional to the current temperature. The subtraction forces the procedure always to accept a true downhill step, whereas the addition randomly leads to accepting an uphill one. Sometimes, it may help the program to fling out from a local minimum. The annealing itself is in decreasing the temperature during the search, T->0. At T=0 the method is exactly the downhill simplex scheme of Nelder and Mead.
Fourier transform

A physical process can be described either in the *time domain*, by the values of some quantity $h$ as a function of time $t$, e.g., $h(t)$, or else in the *frequency domain*, where the process is specified by giving its amplitude $H$ (generally a complex number indicating phase also) as a function of frequency $f$, that is $H(f)$, with $-1 < f < 1$. For many purposes it is useful to think of $h(t)$ and $H(f)$ as being two different representations of the same function. One goes back and forth between these two representations by means of the *Fourier transform* equations,

\[
H(f) = \int_{-\infty}^{\infty} h(t)e^{2\pi ift} \, dt
\]

\[
h(t) = \int_{-\infty}^{\infty} H(f)e^{2\pi iff} \, df
\]

If we use angular frequency, in radians: $\omega = 2\pi f$

\[
H(\omega) = \int_{-\infty}^{\infty} h(t)e^{i\omega t} \, dt
\]

\[h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega)e^{i\omega t} \, d\omega
\]

This is a linear operation.
\[ H(\omega) = \int_{-\infty}^{\infty} h(t)e^{i\omega t} \, dt \]

\( h(t) \) may be considered as coefficients of orthogonal functions \( e^{i\omega t} \) which sum over all values of \( t \) reproduces the function \( H(\omega) \). Therefore, we could find these coefficients by a system of linear equations writing the equation coefficients \( e^{i\omega j t_k} \) for each value of \( t \):

\[
\sum_{k=-\infty}^{\infty} h(t_k)e^{i\omega j t_k} = H(\omega_j)
\]

This is the basis for different indirect methods of Fourier calculations. The Fourier transform may be represented as decomposition of a function to a sum of other orthogonal functions, such as \( \sin, \cos \), etc.
Scattering intensity from a dilute solution of chaotically oriented identical particles is proportional to the spherically averaged scattering from a single particle.

\[ I(s) = 4\pi \int_{0}^{D} p(r) \frac{\sin sr}{sr} \, dr \]

\( p(r) \) is distance distribution function, \( D \) is the maximum particle size.
Case study: computation of the density profiles

The electron density profile of a centrosymmetric bilayer is computed using one-dimensional Fourier transformation as

$$\rho(r) = \sum_{l=1}^{N} s_l A(s, l) \cos(\pi sr) = \sum_{l=1}^{N} s_l [\pm \sqrt{S(l)}] \cos(\pi sr)$$

where $N$ is the number of peaks and $s_l$ and $S(l)$ are the position and the area of the $l$-th peak, respectively. The Lorentz factor $s_l$ reflects random orientation of the lamella microcrystals in PODS. The most plausible combination of signs of the amplitudes $A(s,l)$ is selected by a visual inspection of the density profiles.
PODS samples display up to five Bragg peaks at $s_l=2\pi l/d$, where $d=5.24\pm0.03$ nm is the bilayer thickness.

The long-range order dimension $L=37\pm3$ nm provides the average size of the POSD lamella crystallite (about seven bilayers).
Density profiles of PODS bilayers

- Compound: AuCl₃

- Small particles grow in the central (hydrophilic) part of the bilayer
Literature


