Journal of
Applied Crystallography
ISSN 0021-8898

Received 3 May 02
Accepted 24 July 02

Generic representation and evaluation of properties as a function of position in reciprocal space

Kevin Cowtan

Department of Chemistry, University of York, Heslington, York YO10 5DD, England. Correspondence e-mail: cowtan@yorvic.york.ac.uk

A generalized approach is described for evaluating arbitrary functions of position in reciprocal space. This is a generalization which subsumes a whole range of calculations that form a part of almost every crystallographic software application. Examples include scaling of structure factors, the calculation of structure-factor statistics, and some simple likelihood calculations for a single parameter. The generalized approach has a number of advantages: all these calculations may now be performed by a single software routine which need only be debugged and optimized once; the existing approach of dividing reciprocal space into resolution shells with discontinuities at the boundaries is no longer necessary; the implementation provided makes employing the new functionality extremely simple and concise. The calculation is split into three standard components, for which a number of implementations are provided for different tasks. A 'basis function' describes some function of position in reciprocal space, the shape of which is determined by a small number of parameters. A 'target function' describes the property for which a functional representation is required, for example $\langle|F|^2\rangle$. An 'evaluator' takes a basis and target function and optimizes the parameters of the basis function to fit the target function. Ideally the components should be usable in any combination.

1. Background

Most crystallographic computations depend at some level on statistical properties of either the observations or quantities derived from the observations. This generally involves the calculation of statistical properties of an ensemble of reflections. Depending on the task, this ensemble may include all the reflections for which data are available, or some subset which may be chosen at random [for example, in the calculation of the free R factor (Brünger, 1993)], or systematically on the basis of position in reciprocal space.

The most common example of a calculation using a systematically chosen subset of the data is the calculation of a property over all the reflections in a resolution range, as defined by a spherical shell of a given thickness in reciprocal space. This division is useful because the properties of the diffraction pattern commonly vary as a function of resolution, owing to both the atomic shape (which when averaged over all the atoms in a structure is likely to approach spherical symmetry), and the distribution of interatomic vectors (which also approaches spherical symmetry when averaged). The division of reciprocal space into concentric resolution shells, such that each reflection falls into one shell, is commonly referred to as 'binning', and each shell is referred to as a 'resolution bin'.

The above assumptions concerning spherical symmetry form a partial basis for Wilson's distribution of structure-factor intensities (Wilson, 1949), which describes the expected mean observed intensity as a function of resolution, given some value for the mean thermal motion of the atoms in the structure. This work was extended by Blessing *et al.* (1996) to include a term for the standard deviation of the thermal parameters. To compare this theoretical distribution with the observed data from a real structure, the data are divided into resolution bins and the mean intensity computed over all the reflections in each bin. The mean intensity may then be plotted against the mean resolution of the reflections in the bin to form a distribution of mean intensity as a function of resolution. This curve might then be used to compute the mean thermal motion of the atoms (Wilson, 1949), or to calculate normalized structure factors for use in a direct-methods calculation, such as that of Germain *et al.* (1970).

This approach has some weaknesses. The division of reciprocal space into discrete shells leads to discontinuities across the bin boundaries, and the resolution bins have a smoothing effect which eliminates any variation across the width of the bin. The former may be addressed (incorrectly) by interpolation between bin centres, but the latter is more problematic. To obtain a smoothly varying estimate of a property as a function of resolution, a large number of bins may be required, which means that a small number of reflections will contribute to each bin, giving rise to statistical noise. This is a particular problem in cross-validation calculations, where only a small test set of reflections may be available (Dodson *et al.*, 1996).

An alternative approach is to fit some simple function of resolution to the desired property. For example, Tronrud (1997) uses a function composed of two Gaussians to fit the error in calculated structure factors arising from the omission of the bulk solvent density from the model. Read and Murshudov (Read, 2002; Murshudov *et al.*, 1997) use a four-parameter function of resolution to fit the variation of σ_a as a function of resolution. The problem with these approaches is that the functional form of the curve must be of a sensible form for the particular statistical property, and that a substantial amount of code must be written to implement the function, its derivatives, and a minimizer to determine the optimal values of the parameters.

Zelinka (1998) suggested using a standard family of functions of varying complexity to fit arbitrary statistical properties, in the same way that polynomials of different orders are used to fit a graph to different degrees of accuracy. However, whether a family of functions exists which is suitable for all crystallographic tasks has yet to be seen.

In this work, a generalized method is described for fitting an arbitrary statistical property with an arbitrary function, without the use of resolution bins. The method is sufficiently general to subsume all the approaches described above, and a number of tasks not handled by previous work. This approach may be implemented in a modular fashion using object-oriented programming techniques in such a way that any functional form may be used to fit any statistical property, and a new statistical property, a new functional form, or even a new minimizer may be added with a minimum of effort.

Implementation details are given for several functional forms, including a smooth spline function with particularly desirable properties, and for statistical properties including the *N*th moment of the structure-factor magnitudes and the σ_a function (Read, 1986).

2. A general resolution function evaluator

2.1. Theory

A general resolution function evaluator would allow an arbitrary functional form to be used to represent an arbitrary statistical property of the data. In order to achieve this result using generic reusable components, the problem is divided into three parts. These are as follows.

(i) The 'basis' function. This describes the functional form of the function to be determined, parameterized on a small number of parameters.

(ii) The 'target' function. This describes the statistical property to be fit by the basis function. It is defined such that its value decreases as the fit of the basis function to the desired statistical property improves.

(iii) The resolution function evaluator. This component takes a basis function and a target function and varies the parameters of the basis function to minimize the value of the target function.

These components are described in more detail in the following sections.

2.1.1. The 'basis' function. The basis function defines a value as a function of resolution, or more generally as a function of position in reciprocal space. The function is parameterized on a small number of parameters (typically between 2 and 100). The basis function is therefore a function of both the Miller index $\mathbf{h}$ and the parameters p_i, $i = 1 \ldots n_p$, where n_p is the number of parameters:

$$f_h = f(\mathbf{h}, p_1, \ldots, p_{n_p}). \tag{1}$$

In order for the minimizer to function efficiently, it is also useful to implement the first two derivatives of the basis function with respect to the parameters

$$f'_{\mathbf{h},i} = \left.\frac{\partial f}{\partial p_i}\right|_{p_1,\ldots,p_{n_p}} \tag{2}$$

and

$$f''_{\mathbf{h},i,j} = \left.\frac{\partial^2 f}{\partial p_i \partial p_j}\right|_{p_1,\ldots,p_{n_p}}. \tag{3}$$

The derivatives may be calculated analytically, or by automatic differentiation (see, for example, Griewank *et al.*, 1996), or numerically.

Note that in the common case where f is a function of resolution rather than of general position in reciprocal space, $\mathbf{h}$ becomes $|h|$, computed using either the reciprocal-space metric tensor or a lookup table.

2.1.2. The 'target' function. The target function is a function which is computed over all the reflections and which is minimized by the form of the basis function that gives the best fit to the statistical property for which a functional form is required. Typically, the target function will take the form of a least-squares residual or negative log-likelihood.

In the general case, the target function is a function of all the reflections simultaneously; however, in most practical cases the target function can be written as a sum over all the reflections of some function involving that reflection alone. In this case, the target function can be written as

$$R = \sum_{\mathbf{h}} r_{\mathbf{h}}, \tag{4}$$

where $r_{\mathbf{h}}$ is a function of the data associated with the Miller index $\mathbf{h}$ and of the value of the basis function for that Miller index. For example, if the target function depends on the observed magnitudes, then

$$r_{\mathbf{h}} = r\big[|F(\mathbf{h})|, f_{\mathbf{h}}\big]. \tag{5}$$

The derivatives of $r_{\mathbf{h}}$ with respect to the value of the basis function for that reflection alone may be calculated for use in the minimizer:

$$r'_{\mathbf{h}} = \left.\frac{\mathrm{d}r}{\mathrm{d}f_{\mathbf{h}}}\right|_{f_{\mathbf{h}}} \tag{6}$$

and

$$r''_{\mathbf{h}} = \left.\frac{\mathrm{d}^2 r}{\mathrm{d}f_{\mathbf{h}}^2}\right|_{f_{\mathbf{h}}}. \tag{7}$$

Again, the derivatives may be determined analytically, by automatic differentiation, or numerically.

In some cases, there may be limitations on the valid values of the basis function for a particular target function. 'Bumpers' may be implemented by having the target function routine return 'not-a-number' (NaN) in this case. The minimizer will use this information to adjust the values of the parameters accordingly.

2.1.3. The resolution function evaluator. The resolution function evaluator takes any basis function and target function and varies the parameters of the basis function to determine the functional form that minimizes the value of the target function. It therefore also requires a list of reflections over which the target function is to be evaluated.

In order to optimize the parameters, a Newton–Raphson calculation is performed. This depends on having the first two derivatives of the target function with respect to the parameters, *i.e.* $\partial R/\partial p_i$ and $\partial^2 R/\partial p_i \partial p_j$

Using equations (2) and (6), and the chain rule, the first derivative is obtained as follows:

$$\frac{\partial R}{\partial p_i} = \sum_{\mathbf{h}} \frac{\partial r_{\mathbf{h}}}{\partial p_i} \tag{8}$$

$$= \sum_{\mathbf{h}} \frac{\partial r_{\mathbf{h}}}{\partial f_{\mathbf{h}}} \frac{\partial f_{\mathbf{h}}}{\partial p_i} \tag{9}$$

$$= \sum_{\mathbf{h}} r'_{\mathbf{h}} f'_{\mathbf{h},i}. \tag{10}$$

The second derivative is obtained in the same manner:

$$\frac{\partial^2 R}{\partial p_i \partial p_j} = \sum_{\mathbf{h}} \frac{\partial^2 r_{\mathbf{h}}}{\partial p_i \partial p_j} \tag{11}$$

$$= \sum_{\mathbf{h}} \frac{\partial^2 r_{\mathbf{h}}}{\partial f^2} \frac{\partial f}{\partial p_i} \frac{\partial f}{\partial p_j} + \frac{\partial r_{\mathbf{h}}}{\partial f_{\mathbf{h}}} \frac{\partial^2 f_{\mathbf{h}}}{\partial p_i \partial p_j} \tag{12}$$

$$= \sum_{\mathbf{h}} r''_{\mathbf{h}} f'_{\mathbf{h},i} f'_{\mathbf{h},j} + r'_{\mathbf{h}} f''_{\mathbf{h},i,j}. \tag{13}$$

The Newton–Raphson calculation proceeds by iteratively updating the parameters of the basis function using shifts given by:

$$\delta p_i = -\left(\frac{\partial^2 R}{\partial p_i \partial p_j}\right)^{-1} \left(\frac{\partial R}{\partial p_i}\right). \tag{14}$$

Note that if R is a quadratic function of the parameters, then the calculation converges in a single cycle. This occurs when $r_{\mathbf{h}}$ is a quadratic function of $f_{\mathbf{h}}$ and $f_{\mathbf{h}}$ is linearly dependent on the parameters p_i.

If the target function returns 'NaN', indicating that a bumper has been hit, then the Newton–Raphson step is halved until a valid value is found. A similar procedure is undertaken if the Newton–Raphson step leads to an increase in the value of the target function, unless the dot product of the step and gradient is negative, in which case a gradient step and line search are used instead.

2.2. Implementation

A sample implementation of these ideas has been developed using object-oriented programming techniques as part of the Clipper libraries (Cowtan, 2002), a set of object oriented libraries for the storage and manipulation of all types of crystallographic data. The resulting interface is simple and efficient, with only three C++ instructions being required to optimize the parameters of any basis function using any target function, as follows: one instruction instantiates the chosen basis function; one instruction instantiates the chosen target function; the final instruction calls the evaluator to perform the optimization.

3. Example basis functions

Four basis functions will be considered to illustrate how basis functions may be defined and to test the effectiveness of some simple functional forms. These include a Gaussian function of resolution, an anisotropic Gaussian function, a simple 'binner' which emulates a traditional resolution-bin approach, and a smooth spline function.

3.1. A Gaussian basis function

A Gaussian basis function has only two parameters and so is useful for modelling a property when very few data are available. It is also used in determining the mean temperature factor of the atoms in a structure by finding the Gaussian coefficients required to fit the observed intensity distribution to the theoretical scattering factors for stationary atoms. The equations are as follows:

$$f_{\mathbf{h}} = \exp(-p_1 s + p_0), \tag{15}$$

$$f'_{\mathbf{h},i} = (-s)^i f_{\mathbf{h}} \tag{16}$$

$$f''_{\mathbf{h},i,j} = (-s)^{i+j} f_{\mathbf{h}}, \tag{17}$$

where $s = |h|^2 4 \sin^2\theta/\lambda^2$.

Note that the signs of the parameters are chosen such that for the most common cases the parameters will be positive. The parameter numbering now starts at zero, in line with modern programming practice.

When computing an overall scale and temperature factor, the temperature factor is given by $B = 4p_1$, and the intensity scale factor by $S = \log(p_0)$.

3.2. An anisotropic Gaussian basis function

An anisotropic Gaussian basis function is commonly used to account for anisotropy in the X-ray scattering, which may arise from genuine anisotropy in the atomic motion and disorder, or other effects such as absorption. The basis function has seven parameters: one for scaling and six for the anisotropic coefficients.

For convenience, the equations are defined using a set of intermediate coefficients, c_i:

$$f_{\mathbf{h}} = \exp\left(-\sum_{i=0}^{6} c_i p_i\right), \tag{18}$$

$$f'_{\mathbf{h},i} = -c_i f_{\mathbf{h}}, \tag{19}$$

$$f''_{\mathbf{h},i,j} = c_i c_j f_{\mathbf{h}}, \tag{20}$$

where $c_0 = -1$, $c_1 = -(\mathbf{h}\cdot\mathbf{a}^*)^2$, $c_2 = -(\mathbf{h}\cdot\mathbf{b}^*)^2$, $c_3 = -(\mathbf{h}\cdot\mathbf{c}^*)^2$, $c_4 = -2(\mathbf{h}\cdot\mathbf{a}^*)(\mathbf{h}\cdot\mathbf{b}^*)$, $c_5 = -2(\mathbf{h}\cdot\mathbf{a}^*)(\mathbf{h}\cdot\mathbf{c}^*)$, $c_6 = -2(\mathbf{h}\cdot\mathbf{b}^*)(\mathbf{h}\cdot\mathbf{c}^*)$, where $\mathbf{a}^*$,$\mathbf{b}^*$, $\mathbf{c}^*$ are the vector representations of the reciprocal cell axes. Parameters 1–6 may be converted to anisotropic U values by a scale factor of $2\pi^2$.

3.3. A 'binner' basis function

A basis function can easily be constructed to emulate a traditional 'binner'. This is useful for comparison purposes; however, some minor improvements have also been made over common existing practices.

Traditionally, reciprocal space would be divided into spherical shells, with boundaries determined by raising $|h|$ to some power p, and dividing the range of $|h|^p$ into $n_{\rm bin}$ equal divisions, where p is some power that allows the number of reflections per bin to vary as a function of resolution, and $n_{\rm bin}$ is the number of bins. If $p = 3$, then the number of reflections per bin will be roughly constant if the data are complete. Some calculations benefit from a different distribution, for example the σ_a calculation benefits from having more reflections in the higher resolution bins where σ_a is usually lower and less well determined (Read, 1986).

A better approach allows the problem of incomplete data to be handled. Instead of using the resolution of the reflection, the ordinal number of the reflection in a list sorted by resolution is used. This may be calculated quickly without performing a full sort by calculating a cumulative histogram of the number of reflections as a function of resolution. This may then be used to look up the ordinal from the resolution, using linear interpolation. A power law may also be incorporated in the lookup table. The range of the resulting value may then be divided into equal steps to obtain a bin number for any reflection.

Let $n_{\rm bin}$ be the number of bins and $b(\mathbf{h})$ be the bin number for the Miller index $\mathbf{h}$ calculated as described above. Then the basis function is calculated as follows:

$$f_{\mathbf{h}} = p_{b(\mathbf{h})}, \tag{21}$$

$$f'_{\mathbf{h},i} = \delta[i, b(\mathbf{h})], \tag{22}$$

$$f''_{\mathbf{h},i,j} = 0, \tag{23}$$

where $\delta(i, j) = 1$ if $i = j$, and 0 otherwise.

Note that the binner basis function is linearly dependent on the parameters, and so the matrix of curvatures is the null matrix. The basis function also has compact support, *i.e.* the value of each parameter is only determined by a small subset of the data.

3.4. A spline basis function

The binner basis function is simple and linear, but suffers from the same problems of discontinuity and insensitivity to variations within the bin as traditional 'binner' methods. These limitations can be overcome by using a spline function, *i.e.* a smooth curve, the shape of which is determined by a number of 'control points'.

The quadratic B-spline (Grosse & Hobby, 1994; Cowtan, 1998) is particularly suited to this purpose, since the resulting curve is smooth and continuous, and linearly dependent on the values of the parameters.

The quadratic B-spline is calculated as follows. A set of bin boundaries are calculated using a power law of the reflection ordinal as for the binner above. This time both the bin number $b(\mathbf{h})$ and the fractional position within the bin $\Delta_b(\mathbf{h})$ must be calculated for each Miller index. $\Delta_b(\mathbf{h})$ is represented in the range $-0.5 < \Delta_b(\mathbf{h}) \leq 0.5$, being the position in the bin relative to the bin centre.

The equations for the basis function are then as follows:

$$\begin{aligned} f_{\mathbf{h}} = {} & 0.5[\Delta_b(\mathbf{h}) - 0.5]^2 p_{b(\mathbf{h})-1} + [0.75 - \Delta_b(\mathbf{h})^2] p_{b(\mathbf{h})} \\ & + 0.5[\Delta_b(\mathbf{h}) + 0.5]^2 p_{b(\mathbf{h})+1}, \end{aligned} \tag{24}$$

$$f'_{\mathbf{h},b-1} = 0.5[\Delta_b(\mathbf{h}) - 0.5]^2,$$

$$f'_{\mathbf{h},b} = [0.75 - \Delta_b(\mathbf{h})^2], \tag{25}$$

$$f'_{\mathbf{h},b+1} = 0.5[\Delta_b(\mathbf{h}) + 0.5]^2,$$

$$f''_{\mathbf{h},i,j} = 0, \tag{26}$$

where all other $f'_{\mathbf{h},i}$ are zero. The endpoints of the spline, when $b = 0$ or $b = n_{\rm bin} - 1$ require special treatment. For these points, the parameter beyond the end of the spline is considered to be identical to the endmost value. The derivatives are adjusted accordingly. This has the additional benefit of forcing the gradient of the spline to zero at the origin and the high resolution limit.

Note that as with the binner basis function, the spline basis function is linear and has compact support.

4. Example target functions

Three target functions will be described: one to calculate the nth moment of the structure-factor magnitudes, one to calculate a scale factor for the estimation of normalized structure factors, and the other for the maximum-likelihood refinement of σ_a.

4.1. Target for moments of the structure-factor magnitudes

To define a target function to determine any moment of the structure-factor magnitudes (or any other values) as a function of resolution or position in reciprocal space, a least-squares residual is constructed which minimizes the sum of the squares of the differences between the basis function for each reflection and the required power of the corresponding structure factor.

When calculating moments of structure magnitudes, it is common to use magnitudes which have already been adjusted by removal of the symmetry enhancement factor, *i.e.* $|F_{\rm adj}(\mathbf{h})|^2 = |F(\mathbf{h})|^2/\varepsilon(\mathbf{h})$, where $\varepsilon(\mathbf{h})$ is given by the number of symmetry operators divided by the number of symmetry-equivalent copies of the reflection in the hemisphere.

Let n be the required moment of the structure-factor magnitudes. Then the equations for the target function are as follows:

$$r_{\mathbf{h}} = \left[f_{\mathbf{h}} - |F_{\rm adj}(\mathbf{h})|^n\right]^2, \tag{27}$$

$$r'_{\mathbf{h}} = 2\left[f_{\mathbf{h}} - |F_{\rm adj}(\mathbf{h})|^n\right], \tag{28}$$

$$r''_{\mathbf{h}} = 2. \tag{29}$$

At the minimum of the target function, the gradient term $r'_{\mathbf{h}}$ will be equal to zero. If the basis function $f_{\mathbf{h}}$ is a simple binner, then this equation reduces to the equation for the nth moment of a set of structure-factor magnitudes.

Since this target function is a quadratic function of the basis functions, then if the basis function is linear, the target function will be a quadratic function of the basis function parameters. In this case, the Newton–Raphson calculation will converge in a single cycle and the whole calculation will not be significantly more demanding than a conventional calculation using resolution bins.

4.2. Target for data scaling

Normalized structure factors, or E's, are commonly calculated by scaling the data in resolution shells to fit the identity $\langle|E(\mathbf{h})|^2\rangle = 1$. When performing a calculation in resolution shells, this may be achieved by simply dividing all the adjusted magnitudes in that shell by the mean of the squared adjusted magnitude in that shell, which could be calculated using the target function described above. However, when using more complex basis functions, this approach is invalid, as the functional fit to the reciprocal of a function is different to the reciprocal of the functional fit to that function. Therefore, a different target function is needed for this problem. The appropriate form of the target function is as follows:

$$r_{\mathbf{h}} = \left[f_{\mathbf{h}}|F_{\rm adj}(\mathbf{h})|^2 - 1.0\right]^2/|F_{\rm adj}(\mathbf{h})|^2, \tag{30}$$

$$r'_{\mathbf{h}} = 2\left[f_{\mathbf{h}}|F_{\rm adj}(\mathbf{h})|^2 - 1.0\right], \tag{31}$$

$$r''_{\mathbf{h}} = 2|F_{\rm adj}(\mathbf{h})|^2. \tag{32}$$

The same target function, with 1.0 replaced by the expected scattering intensity as a function of resolution, can be used in conjunction with a Gaussian basis function to determine the overall temperature factor of a data set.

4.3. Target for σ_a refinement

The variable σ_a is used to estimate what fraction of a calculated normalized structure factor $E_c(\mathbf{h})$ is correct, given a set of observed structure-factor magnitudes $|E_o(\mathbf{h})|$. This information may then be used to calculate phase probability distributions and map coefficients.

A σ_a target may be implemented by defining a target function which is the negative of the log-likelihood function, using the likelihood expression of Murshudov *et al.* (1997, equation 17 therein), or Read (1986, equation A1 therein). Minimizing this function corresponds to maximizing the overall likelihood.

It is possible to choose the variable to be fit by the basis function in any way that is convenient to the problem. While it has been traditional to refine σ_a directly, Read (2002) suggested an alternative approach: the basis function is instead used to define the value $\omega_a = \sigma_a/(1-\sigma_a^2)$. The likelihood as a function of this parameter is closer to quadratic, and so convergence is better and the expressions are simpler. In addition, ω_a is only bounded at zero rather than between zero and one, reducing the need for the 'bumpers' described in §§2.1.2 and 2.1.3. Refinement of both σ_a and ω_a will be considered; the extension to other parameterizations is trivial.

The target function and its derivatives with respect to ω_a were derived from the work of Murshudov *et al.* (1997), using the assumption that all occurrences of $|E|^2$ will average out to 1. This assumption is strictly valid as long as the basis function is linear and the same basis function is used for normalization and σ_a evaluation; this has been confirmed in practice.

Omitting a constant and scale factor, the log-likelihood and its derivatives with respect to ω_a are as follow. For centric reflections:

$$\mathrm{LLK}_{\mathbf{h}} = \frac{1}{1-\sigma_a^2} + \frac{1}{2\log(1-\sigma_a^2)} - \log[\cosh(X/2)], \tag{33}$$

$$\frac{\partial \mathrm{LLK}_{\mathbf{h}}}{\partial \omega_a} = \sigma_a - |E_o||E_c|\tanh(X/2), \tag{34}$$

$$\frac{\partial^2 \mathrm{LLK}_{\mathbf{h}}}{\partial \omega_a^2} = \frac{(1-\sigma_a^2)^2}{(1+\sigma_a^2)} - |E_o|^2|E_c|^2[1-\tanh(X/2)^2], \tag{35}$$

where $\sigma_a = [(4\omega_a^2+1)^{1/2}-1]/2\omega_a$ and $X = 2|E_o||E_c|\omega_a$.

For acentric reflections:

$$\mathrm{LLK}_{\mathbf{h}} = \frac{2}{1-\sigma_a^2} + \log(1-\sigma_a^2) - \log[I_0(X)], \tag{36}$$

$$\frac{\partial \mathrm{LLK}_{\mathbf{h}}}{\partial \omega_a} = 2\sigma_a - 2|E_o||E_c|\,\mathrm{sim}(X), \tag{37}$$

$$\frac{\partial^2 \mathrm{LLK}_{\mathbf{h}}}{\partial \omega_a^2} = 2\frac{(1-\sigma_a^2)^2}{(1+\sigma_a^2)} - 4|E_o|^2|E_c|^2\frac{\partial[\mathrm{sim}(X)]}{\partial X}, \tag{38}$$

where $\mathrm{sim}(X) = I_1(X)/I_0(X)$.

To form the target function to refine a basis function fit to ω_a, these expressions are used directly:

$$r_{\mathbf{h}} = \mathrm{LLK}_{\mathbf{h}}, \tag{39}$$

$$r'_{\mathbf{h}} = \frac{\partial \mathrm{LLK}_{\mathbf{h}}}{\partial \omega_a}, \tag{40}$$

$$r''_{\mathbf{h}} = \frac{\partial^2 \mathrm{LLK}_{\mathbf{h}}}{\partial \omega_a^2}. \quad (41)$$

This target function can be reliably used to calculate ω_a and therefore σ_a in resolution shells by using the binner-emulator basis function. Convergence is extremely good; thus the initialization step used by Read (1986) to initialize σ_a to some sensible value in each resolution shell is not required. Unfortunately, this approach is not always reliable when using the spline basis function, because the large dynamic range of ω_a leads to oscillations when attempting to fit a smooth basis function.

To form the target function to refine a basis function fit to σ_a, the derivatives are adjusted accordingly:

$$r_{\mathbf{h}} = \mathrm{LLK}_{\mathbf{h}}, \quad (42)$$

$$r'_{\mathbf{h}} = \frac{\partial \mathrm{LLK}_{\mathbf{h}}}{\partial \omega_a}\frac{\partial \omega_a}{\partial \sigma_a}, \quad (43)$$

$$r''_{\mathbf{h}} = \frac{\partial^2 \mathrm{LLK}_{\mathbf{h}}}{\partial \omega_a^2}\left(\frac{\partial \omega_a}{\partial \sigma_a}\right)^2 + \frac{\partial \mathrm{LLK}_{\mathbf{h}}}{\partial \omega_a}\frac{\partial^2 \omega_a}{\partial \sigma_a^2}. \quad (44)$$

This function behaves in a similar manner to the original approach of Read (1986) when used in resolution shells. It can be used with a spline basis function and leads to a sensible result in some cases, but convergence is generally poor and often fails completely.

Further work is required to find a universal combination of smooth basis function and target function for σ_a evaluation. The problems seem to arise because σ_a is very poorly defined for small batches of reflections, so any smooth function of resolution will try and fit sharp fluctuations in σ_a across small ranges of resolution. The traditional calculation in resolution shells imposes an artificial smoothing effect which eliminates this problem.

5. Selection of initial values for parameters

Initial values are required from which to start the refinement of the basis-function parameters. A sensible choice of these parameters depends on the combination of basis and target function to be used. However, in the case of a linear basis function and quadratic target function, any starting values lead to convergence in a single cycle, so it is normal to start with all the parameters at zero.

In the general case, an elegant method is available to remove the effect of the choice of basis function. A new data list is generated, containing a floating-point value for each reflection, initialized to some suitable value. The selected basis function is then used to fit this function, using the first-moment target function (which converges in a single cycle from any starting point). The resulting parameters may be used to initialize the subsequent calculation.

In the case of σ_a calculation, Read (1986) demonstrated that σ_a may be estimated from the square-root of the correlation between the observed and calculated squared magnitudes. This in turn may be calculated by using the resolution function evaluator to calculate the required moments of $|E_o|$,$|E_c|$ and $|E_o||E_c|$ as a function of resolution. However, if the ω_a target function for σ_a is used, convergence is sufficiently good that it is sufficient to start the calculation with $\omega_a(|\mathbf{h}|) = 1.0$.

6. Optimization

Optimization of this approach allows computational efficiencies that are comparable with a traditional calculation in resolution shells. The optimization is performed by implementing 'hints' in each basis and target function which allow the evaluator to avoid any unnecessary steps. The hints are as follows.

(i) Each basis function can specify the number of significant diagonals in the upper triangle of its curvature matrix. For the 'binner' basis function, this value is 1, indicating a diagonal matrix; for the 'spline' basis function, this value is 3, indicating a pentadiagonal matrix. Only the non-zero values are then used in constructing the curvature matrix for the target function.

(ii) Each basis function can specify whether it is 'LINEAR', indicating that the value of the function is linearly dependent on the parameters, or 'GENERAL'.

(iii) Each target function can specify whether it is 'QUADRATIC', indicating that the value of the function is quadratically dependent on the value of the basis function for any reflection, or 'GENERAL'.

If the basis function is 'LINEAR' and the target function is 'QUADRATIC', then the Newton–Raphson calculation is guaranteed to converge in a single step, and so any further calculation can be omitted.

7. Results

The 'Clipper' implementation was used to demonstrate this approach. The convenience of the method is clear, with only three lines of C++ code required for each calculation. In the examples considered below, the possible benefits of fitting continuous functions of position in reciprocal space will be examined.

7.1. Structure-factor statistics

The variation in the mean scattering intensity as a function of resolution, $\langle |F|^2 \rangle$, is commonly calculated as the basis of a Wilson plot, or as part of a scaling calculation (but see §4.2). It is interesting to see how the fit of this function may be improved by fitting a function to the data, as opposed to calculating averages over resolution shells.

To test the effect of different basis functions, each basis function was used with the 'moments' target (§4.1) to fit the data. In the cases of basis functions with variable numbers of parameters (*i.e.* the binner and spline), the number of parameters was varied over the range 1–25.

The quality of the estimate of $\langle |F|^2 \rangle$ was tested by using a full cross-validation calculation (Brünger, 1993) with 20 free sets; *i.e.* the data were divided into 20 sets. Each set was chosen

as a free set in turn, and the remaining 19 sets used to calculate the resolution function. The match between the resulting resolution function and the data was then tested using the following free-R-factor like quantity:

$$R_{\rm free} = \frac{\sum_{h\in{\rm free}} w_h\left[|F(\mathbf{h})|^2 - f(\mathbf{h})\right]^2}{\sum_{h\in{\rm free}} w_h\left[|F(\mathbf{h})|^2\right]^2}, \qquad (45)$$

where w_h corrects for the calculation over a reciprocal asymmetric unit only. (Similar results were also obtained by calculating a correlation over F or E values.) The calculation was repeated for each free set in turn, and the resulting statistics combined to reduce the impact of noise on the results.

The test data came from the GerE data (Ducros *et al.*, 2001) distributed with the *CCP4* suite (Collaborative Computational Project, Number 4, 1994). The data were truncated to 3.0 Å resolution for the purposes of these calculations.

The values of the cross-validated residuals are shown in Fig. 1 for the Gaussian basis function, and as a function of the number of parameters for the binner and spline basis functions. Two lines are shown each for the binner and spline functions, the first with a linear distribution of bins or control points (*i.e.* equal numbers of reflections per bin), and the second for a quadratic distribution of bins or control points (*i.e.* more bins at low resolution and more reflections per bin at high resolution).

Note that the spline basis function is always better than the binner. Also, the quadratic distribution of bins or control points is dramatically better than the linear distribution, because it provides a better model of the low-resolution region of the diffraction pattern where there are fewer reflections and the mean intensity varies most rapidly. (The quadratic distribution is not always practical, since many calculations will require some fixed minimum observation/parameter ratio over the whole data set.) A Gaussian provides the best fit when only two parameters are available, but a binner or spline with a quadratic distribution of control points is better than the exponential polynomial function for four parameters.

The graph shows some other interesting features. The optimal number of spline control points when they are linearly spaced is 22. When the control points are quadratically spaced, the optimum number is only 10. Beyond these numbers, the free residual increases, indicating that over-fitting is occurring. The results for the corresponding binner vary erratically with the number of parameters, as the bin boundaries move into or out of phase with the features of the Wilson plot.

Since the features of the Wilson plot for a protein are to a first approximation independent of the structure (Cowtan, 1998), it seems likely that there is an optimal distribution of control points where the positions of the control points are adjusted to the features of the Wilson plot.

The corresponding plot of $\langle|F|^2\rangle$ as a function of resolution, showing some examples of the spline and binner estimates, is shown in Fig. 2. Note that beyond 10 control points, the features of the spline curve do not change significantly.

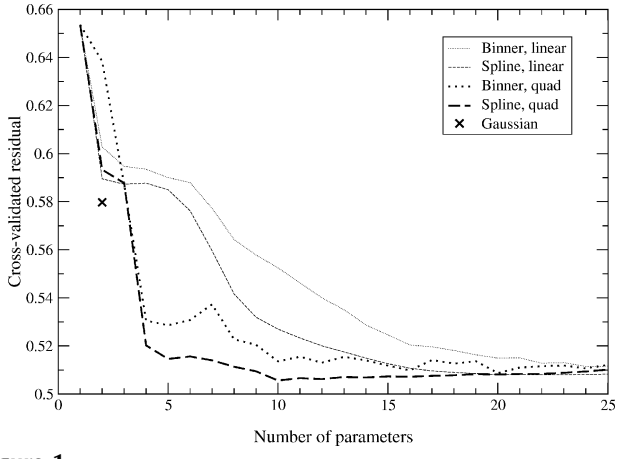

Figure 1
Cross-validated residual between $|F|^2$ and $\langle|F|^2\rangle$ as a function of the number of parameters for six basis functions: binner with linear resolution scale; spline with linear resolution scale; binner with quadratic resolution scale; spline with quadratic resolution scale; Gaussian; exponential polynomial.

7.2. Anisotropic scaling

To test the anisotropic Gaussian basis function, the basis function was used with the modified scaling function described at the end of §4.2 to calculate an overall anisotropic displacement parameter ($U_{\rm aniso}$) for a model.

The test data were provided by synthetic structure factors calculated from the well known P1 lysozyme data (Walsh *et al.*, 1998). To isolate the effects of anisotropic variation in the diffraction pattern due to the effects of sampling in reciprocal space from the overall anisotropy, two sets of structure factors were calculated, using first the true $U_{\rm aniso}$ for each atom, and secondly the modified $U'_{\rm aniso}$, which differed from the true values by the addition of a constant anisotropic factor to the values for each atom. The mean squared form factors were fitted to the squared magnitude data over the resolution range 3.5–1.5 Å, and the results divided by two to give an overall $U_{\rm aniso}$.

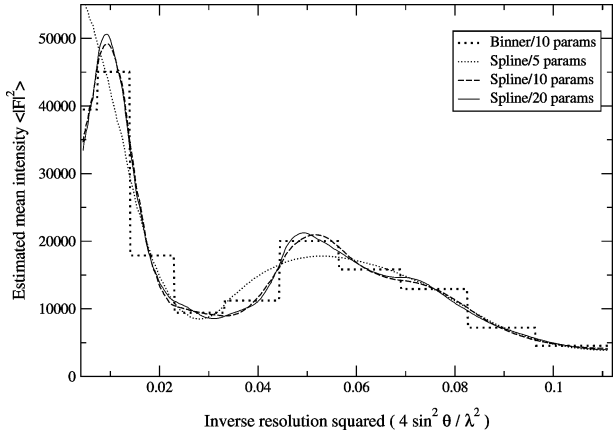

Figure 2
Graph of $\langle|F|^2\rangle$ against resolution using binner with 10 bins and splines with 5, 10 and 15 control points.

Table 1
Overall $U_{\rm aniso}$ for true and modified lysozyme models between 35 Å and 15 Å.

The difference in $U_{\rm aniso}$ closely fits the change to the model.

	U_{11}	U_{22}	U_{33}	U_{12}	U_{13}	U_{23}
Overall $U_{\rm aniso}$ (true model)	0.1009	0.0988	0.1202	0.0020	0.0019	0.0063
Overall $U'_{\rm aniso}$ (modified model)	0.1972	0.0963	0.1678	0.0517	0.0268	0.0053
Difference $U_{\rm aniso} - U'_{\rm aniso}$	0.0963	−0.0025	0.0476	0.0497	0.0250	−0.0010
Applied shift	0.1000	0.0000	0.0500	0.0500	0.0250	0.0000

The results are tabulated in Table 1. The difference between the overall $U_{\rm aniso}$ for the initial and modified structures closely corresponds to the change applied to the atomic $U_{\rm aniso}$.

7.3. σ_a estimation

The use of the spline basis function in the estimation of σ_a was also investigated. The stability problems that occur with the current parameterization and optimizer limited the flexibility of these tests. A range of synthetic test data sets were generated, for some of which the calculation of the spline fit to σ_a did not converge. A representative example for which convergence was achieved was selected for further examination.

The GerE data set, from the previous section, was used in the tests. The refined model was used to calculate a set of 'ideal' phases, using *REFMAC* version 5 with the bulk-solvent correction enabled (Murshudov *et al.*, 1997). Then the solvent atoms and 20% of the protein atoms were removed, and the remaining atoms were randomized by adding an independent random coordinate error of 0.5 Å to each atom. (While unrealistic, this is consistent with the assumptions of the σ_a calculation.) New structure-factor magnitudes and phases were then calculated for the truncated model, again using *REFMAC*.

The σ_a calculation was then performed, both using resolution shells and the resolution spline function for both the scaling and the σ_a evaluation. Each calculation was performed both for 6 and for 12 parameters (*i.e.* resolution bins or control points). (Note that for these test data both the σ_a and ω_a target functions converged and produced almost identical results.)

Figures-of-merit for the synthetic phases were calculated using the σ_a estimates obtained by the various methods in the same way as Read (1986), except that a different estimate of σ_a is available for every single reflection. These figures-of-merit were compared with the actual value of the cosine of the phase error *versus* the refined structure by calculating a linear regression of FOM (from σ_a) against $\cos(\Delta\varphi)$. The results are given in Table 2.

The ideal values are $m = 1$ and $c = 0$; these are not achieved owing to the limitations of the protein model. It can be seen that with 12 parameters, both methods perform similarly. When the number of parameters is reduced to 6, the spline continues to give very similar results, but the results of the binner are degraded.

Table 2
Linear regression coefficients m/c fitting FOM = $m\cos(\Delta\varphi) + c$, for binner and spline calculations with 6 and 12 parameters.

m/c	Binner	Spline
6 parameters	0.829/0.072	0.848/0.060
12 parameters	0.847/0.059	0.850/0.056

Maps calculated using figures-of-merit derived from the various σ_a estimates are visually indistinguishable, and the differences in correlations with a true map are not significant. However, the initial indications suggest that the approach is valid.

8. Future development

There are a number of other ways in which this approach may be developed, as follow.

(i) More sophisticated minimization procedures, employing, for example, singular value decomposition, may improve convergence when handling likelihood functions. Alternatively, re-parameterizing the problem may be effective. Once this is solved, the problem of calculating σ_a from small cross-validation test sets may be investigated.

(ii) New basis functions and target functions can be added. For a specific task, these may be added as part of an application, but there may be other functions that are of general use to many applications which should be added to the library.

(iii) The technique may be extended to handle target functions in which the reflections are not independent. These can be computed in the current scheme if the second derivative matrix of the target function R is diagonal dominant, in which case the calculation described here will work as a diagonal approximation to the full Newton–Raphson calculation.

(iv) The technique may be extended to handle functions of several variables. For example, Murshudov (2002) suggests the separate refinement of σ_a and D as part of an improved likelihood calculation for refinement and map calculation.

9. Conclusions

A general scheme has been described for evaluating statistical properties of the data as a function of position in reciprocal space. This generalizes a wide range of existing calculations. As a result, all of those calculations may now be performed by a single set of routines, requiring a minimum amount of problem-specific code. This increases productivity for the developer and reduces support overhead. The application interface is particularly convenient and concise. The generality of the approach also allows the use of more sophisticated functions of position in reciprocal space, for which the resolution spline function seems particularly suitable.

The method has been demonstrated for the calculation of a simple structure-factor statistic, for which superior results were obtained through the use of smooth basis functions. The same code has been demonstrated for σ_a estimation, although

currently this is only reliable when using the 'binner' emulator and so the only benefit is convenience. It is hoped that this limitation will eventually be removed by the use of alternative parameterizations or optimization methods.

The author would like to thanks R. Read and G. Murshudov for their helpful suggestions in the implementation of the σ_a target function. This work was funded by The Royal Society grant number 003R05674.

References

Blessing, R. H., Guo, D. Y. & Langs, D. A. (1996). *Acta Cryst.* D**52**, 257–266.
Brünger, A. T. (1993). *Acta Cryst.* D**49**, 24–36.
Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* D**50**, 760–763.
Cowtan, K. D. (1998). *Acta Cryst.* D**54**, 487–493.
Cowtan, K. D. (2002). *Joint CCP4 and ESF-EACBM Newsletter on Protein Crystallography*, **40**.
Dodson, E. E., Kleywegt, G. J. & Wilson, K. S. (1996). *Acta Cryst.* D**52**, 228–234.
Ducros, V. M. A., Lewis, R. J., Verma, C. S., Dodson, E. J., Leonard, G., Turkenburg, J. P., Murshudov, G. N., Wilkinson, A. J. & Brannigan, J. A. (2001). *J. Mol. Biol.* **306**, 759–771.
Germain, G., Main, P. & Woolfson, M. M. (1970). *Acta Cryst.* B**26**, 274–285.
Griewank, A., Juedes, D., Mitev, H., Utke, J., Vogel, O. & Walther, A. (1996). *ACM Trans. Math. Software*, **22**, 131–167.
Grosse, E. & Hobby, J. D. (1994). *Math. Comput.* **63**, 175–194.
Murshudov, G. N. (2002). *Estimation of Overall Parameters of Likelihood Function*, http://www.ysbl.york.ac.uk/~garib/ml_sigma/ml_sigma.html.
Murshudov, G. N., Vagin, A. A. & Dodson, E. J. (1997). *Acta Cryst.* D**53**, 240–255.
Read, R. J. (1986). *Acta Cryst.* A**42**, 140–149.
Read, R. J. (2002). Personal communcation.
Tronrud, D. (1997). *Methods Enzymol.* **277B**, 306–319.
Walsh, M. A., Schneider, T. R., Sieker, L. C., Dauter, Z., Lamzin, V. S. & Wilson, K. S. (1998). *Acta Cryst.* D**54**, 522–546.
Wilson, A. J. C. (1949). *Acta Cryst.* **2**, 318–321.
Zelinka, J. (1998). Personal communcation.