Acta Cryst. (1997). D**53**, 371–376

Bias Reduction in Phase Refinement by Modified Interference Functions: Introducing the γ Correction

JAN PIETER ABRAHAMS

MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, England. E-mail: jpa@mrc-lmb.cam.ac.uk

(Received 12 September 1996; accepted 9 December 1996)

Abstract

The chemical, physical and symmetry constraints of an electron-density map impose relationships between structure factors, and these relationships are exploited during refinement. However, constraints often allow an artificially high correlation between the model and the original structure factors, a flaw known as model or refinement bias. Elimination of the bias component of a constrained model, the component insensitive to constraints, enhances the power of phase-refinement techniques. The scale of the bias component, here denoted as γ, is shown to be equal in magnitude to the origin vector of the interference function **G** that defines the relationships between the structure factors. The γ correction leads to solvent flipping in the case of phase improvement by solvent flattening, and other types of constraint allow a similar treatment.

1. Model bias

Since only amplitudes of structure factors can be measured directly, the phases of electron-density maps need to be inferred or refined from a model. Such a model is not necessarily just an atomic model. It might include a description of non-crystallographic symmetry, for example, and also solvent flatness is an implicit, but very important aspect of atomic models. Obviously, only a correct model will provide proper phases, and its correctness is usually estimated by a comparison between its structure-factor amplitudes and the observed ones.

Model bias is the tendency of a calculated electron-density map to confirm the model that provided its phases, even if the model is only partially correct. In such cases, σ_A-weighted structure-factor amplitudes minimize model bias (Srinivasan & Ramachandran, 1965; Read, 1986). However, an underlying assumption of this weighting scheme is that the model and the observed data are mutually independent. Usually this is not the case if a model has been refined, because then the model has already been adapted to have structure-factor amplitudes more like the experimentally observed ones. After refinement, density maps will therefore be biased towards the original starting map, even if σ_A weighting is used. This causes premature convergence to a wrong solution, with model structure-factor amplitudes agreeing with experimental ones, yet being phased incorrectly. It is for this reason that structure solutions need to be validated.

A powerful tool for the validation of structural models is the free R factor (Brünger, 1992). It is a measure of the differences between model structure-factor amplitudes and a subset of observed structure-factor amplitudes not used in the refinement. If the final model is correct, this should result in the correct reconstruction of the subset.

In the early stages of a structure solution, when interpretation of electron density in terms of an atomic model requires the improvement of phases, constraints need to be derived from general criteria. The following criteria have been used: solvent flatness (Wang, 1985), non-crystallographic symmetry (Bricogne, 1974), the general atomic shape described by Sayre's equation (Zhang & Main, 1990*a*), and the expected histogram of the electron density (Zhang & Main, 1990*b*). However, models constructed from these constraints are less restrictive than atomic models, and their validation requires a more elaborate approach than the calculation of a free R factor. One of the problems in these cases is that leaving out a subset of structure factors can greatly influence the maps used for generating the model, thus reducing the power of the technique (Grimes & Stuart, 1994) – without due care and attention one could end up with validated, but sub-optimal electron density.

Complete validation avoids this problem. Here, the data are divided into ten to 20 different subsets and a series of density maps is calculated, as each of these subsets is left out in turn. After modification of the maps, only the reconstructed structure factors are used for the calculation of the final map (Cowtan & Main, 1996). Below, a procedure is described which allows an economic and straightforward implementation of the most rigorous complete validation. It is comparable to leaving out each observation in turn in a single round of refinement, and is especially suited for phase refinement without an atomic model.

2. The interference function G

The convolution of two functions at a single point is determined by shifting the origin of the first function to this point, and then calculating the integral of the product of the two functions. This procedure is repeated

for each point of the second function, every time shifting the origin of the first. For example, the convolution of a perfect image with a response function that describes how an imperfect measuring device smears out a single pixel, will result in the blurred image that would be measured by this device.

Convolution can be a tedious and lengthy process. However, according to the convolution theorem, there is a more economical way to obtain the same result. This theorem states that the Fourier transform of a convolution of two functions is equal to the product of the individual Fourier transforms of each of the two functions.

For the sake of clarity, a shorthand notation will be used throughout the rest of the paper, which will be introduced here. The Fourier transform within a unique volume V of a function $\mathbf{f}$ is equal to $\mathbf{F}$, where $\mathbf{f}$ and $\mathbf{F}$ are functions of a vector $\mathbf{x}$,

$$\mathbf{F} \equiv F(\mathbf{y}) = \int_V f(x)\exp(2\pi i x y)\delta^3 x,$$

$$\mathbf{f} \equiv f(x) = V^{-1}\int F(\mathbf{y})\exp(-2\pi \mathbf{i}\mathbf{x}\mathbf{y})\delta^3 \mathbf{y}.$$

Convolution is indicated by the symbol *

$$\mathbf{F}*\mathbf{G} \equiv (F*G)(\mathbf{x}) = \int F(\mathbf{y})G(\mathbf{x}-\mathbf{y})\delta\mathbf{y}.$$

If the symbol $\Leftrightarrow$ denotes a Fourier pair as in $\mathbf{F} \Leftrightarrow \mathbf{f}$ and $\mathbf{G} \Leftrightarrow \mathbf{g}$, then the convolution theorem can be formulated as follows,

$$\mathbf{F}*\mathbf{G} \Leftrightarrow \mathbf{fg}. \qquad (1)$$

In protein crystallography, a real-space shape function $\mathbf{g}$ is usually defined by grid points taking on either the value '1' or '0'. The Fourier transform $\mathbf{G}$ of the shape function is often referred to as the interference function (Rossmann & Blow, 1962).

3. $\mathbf{G}_\gamma$ and solvent flattening

Density modification by solvent flattening can be described as the multiplication of a shape function with an electron-density map. Here, the shape function describes the protein mask, where grid points corresponding to protein are set to '1', whilst those within the solvent region are set to '0'. If a density map has been calculated with an $F000$ of zero, and if subsequently the difference between the solvent and protein electron density is added to every grid point, a correctly phased map will remain unchanged upon multiplication with the shape function,

$$\mathbf{g}(\mathbf{c}+\mathbf{f}) = \mathbf{c}+\mathbf{f}. \qquad (2)$$

Here, $\mathbf{c}$ is a constant function, $\mathbf{g}$ is the shape function of the protein, $\mathbf{f}$ is the correct electron-density map.

Although computationally very inefficient, it is useful to consider the effects of solvent flattening in reciprocal space, where the Fourier transform of the shape function describes the dependence of each of the individual structure factors on itself and surrounding ones (*e.g.* Bricogne, 1974). Provided the phases are assigned correctly, none of the structure factors will be affected by the convolution with the interference function. This becomes apparent from the Fourier transform of (2) [see also Crowther (1967), note that the Fourier transform $\mathbf{C}$ of the constant function $\mathbf{c}$ is zero everywhere except at the origin]

$$\mathbf{G}*(\mathbf{F}+\mathbf{C}) = \mathbf{F}+\mathbf{C}. \qquad (3)$$

A simple multiplication by zero in real space is equivalent to a series of vector multiplications and additions in reciprocal space, in all cases resulting in identity! See also Fig. 1. Solvent flatness, therefore, poses powerful constraints on structure factors, especially when the solvent content is high, and this merits a more detailed examination of the associated interference function.

Fig. 2 shows the radial distribution of the intensity of the interference function $\mathbf{G}$ corresponding to the Fourier transform of a 3.2 Å mask of the protein complex F_1-ATPase (Abrahams, Lutter, Leslie & Walker, 1994). Most of the intensity of this interference function is around the origin. Upon convoluting the structure factors $\mathbf{F}$ with $\mathbf{G}$, structure factors will therefore mainly be recombined locally in reciprocal space.

The convolution with $\mathbf{G}$ improves phases of incorrectly phased structure factors because the many vector additions cancel out random errors, thus enhancing systematic correlations between structure factors. A better map results, and after recombining the σ_A-weighted observed structure-factor amplitudes with the phases of the improved map and possibly with experimental phase information, the procedure can be iterated. Provided every iteration results in a better map, eventually the correct map might be produced when the condition expressed in (2) is reached. However, in practice this turns out not to be the case. Already in the very first iterations, errors that are not cancelled out by the convolution will compromise the procedure.

One clear source of systematic errors is the phase recombination, where the independence between the observed structure-factor amplitudes and the convoluted structure-factor amplitudes has to be assumed. Where this assumption is incorrect, the quality of the convoluted structure factors will be overestimated, resulting in a bias.

Since the shape function $\mathbf{g}$ in (2) is real, its Fourier transform $\mathbf{G}$ in (3) is symmetrical. Although $\mathbf{G}$ is a complex function, the origin vector will never have an imaginary component because of its symmetry, and the magnitude of the origin vector, denoted here by the scalar γ, will reflect the mean value of $\mathbf{g}$. Consider the convolution $\mathbf{G}*\mathbf{F}$ as consisting of two components (see

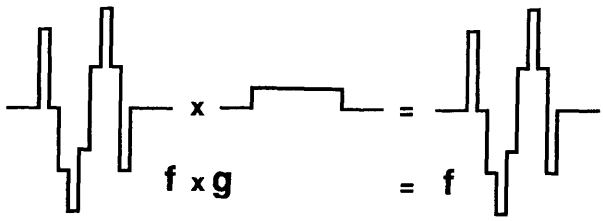
also Fig. 1*b*)

$$\mathbf{G}^*\mathbf{F} = \mathbf{G}_\gamma{}^*\mathbf{F} + \gamma\mathbf{F}, \quad (4)$$

where $\mathbf{G}_\gamma$ is the interference function **G** with its origin vector set to zero, and where γ is the magnitude of the origin vector of **G**.†

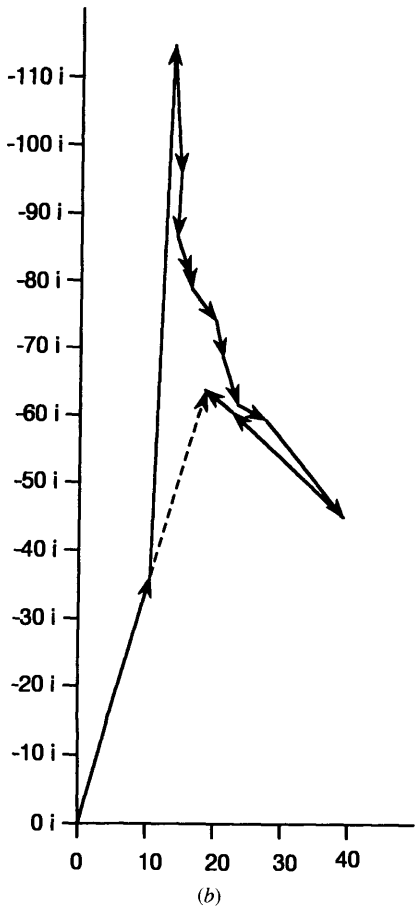
† Equation (4) follows directly from the convolution theorem: $\mathbf{G}^*\mathbf{F} = (\mathbf{G}_\gamma + \mathbf{D})^*\mathbf{F} = \mathbf{G}_\gamma{}^*\mathbf{F} + \mathbf{D}^*\mathbf{F} = \mathbf{G}_\gamma{}^*\mathbf{F} + \gamma\mathbf{F}$. (Where $\mathbf{G}_\gamma$ is defined as above and **D** is a Dirac function which is zero exept at the origin, where it equals the origin of **G**. From the definition of the convolution it follows that **F*****D** is equal to a scale factor times **F** if **D** is a Dirac function. The scale factor is determined by the the value of the origin of **D**.)

This equation forms the foundation on which the major conclusions of this paper are based. It allows separation of the information provided by the shape function ($\mathbf{G}_\gamma{}^*\mathbf{F}$) from the information that was already present in the original structure factors ($\gamma\mathbf{F}$). It implies that the result of a convolution of a data set of structure factors with an interference function is biased towards the original data set by a factor γ. Even a structure factor in complete disagreement with the rest of the data will contain a component of its original value after the convolution. Given a convolution **G*****F**, the bias can trivially be removed when γ is known. All that is required is to subtract the original, unconvoluted data



f : 0, 0, 0, 4, 0, -3, -5, -2, 2, 5, 2, -3, 0, 0, 0, 0
g: 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0

(*a*)



(*b*)

Fig. 1. (*a*) When two functions **f** and **g** are multiplied in real space, the product can be identical to **f**. For example, this is the case when **g** is '0' where **f** is '0', but where **g** is '1' otherwise. It follows from the convolution theorem that if (**fg** = **f**), then the convolution of the Fourier transforms of **f** and **g** (here denoted as **F** and **G**, respectively) must equal the Fourier transform of **f**: (**fg** = **f**⇔**F*****G** = **F**). An example is given in a one-dimensional case. Here the Fourier transform **F** of **f** is: (0), (1.177–3.933*i*), (5.536 + 14.778i), (–14.508–12.948*i*), (5 + 3*i*), (6.266–3.481*i*), (–1.536 + 0.778*i*), (–0.934 + 5.966*i*), (–2), (-0.934-5.966*i*), (–1.536–0.778*i*), (6.266 + 3.048*i*), (5–3*i*), (–14.508 + 12.947*i*), (5.536-14.778*i*), (1.765 + 3.933*i*). The Fourier transform **G** of **g** is: (9), (–4.645 + 1.924*i*), (–0.707 + 0.707*i*), (0.572–1.383*i*), (–*i*), (0.256 + 0.617*i*), (0.707 + 0.707*i*), (–0.184–0.076*i*), (–1), (–0.184 + 0.076*i*), (0.707–0.707*i*), (0.256–0.617*i*), (*i*), (0.573 + 1.383*i*), (–0.707–0.707*i*), (–4.645–1.924*i*). (*b*) The identity **F*****G** = **F** is demonstrated for the vector with index 1 [which is (1.177 – 3.933i) in **F**]. In order to determine the new vector after convolution, 16 vector multiplications are required, after which the sum of the products is calculated. The graph shows the summation of the vector products in the complex plane as a dashed line which starts at the origin. Note that the sum is (18.83–62.93*i*), which, after scaling down by a factor of 16 (the number of data points in the convoluted function), equals (1.177–3.933*i*). Also note that the vector depicted by a solid arrow starting at the origin, and which coincides with the summation of all vectors, is the product of the original vector and the origin vector of **G**.

set, scaled by γ,

$$\mathbf{G}\gamma^*\mathbf{F} = \mathbf{G}^*\mathbf{F} - \gamma\mathbf{F}. \quad (4a)$$

Substitution of (4) into (3) shows that, given correct phases, the removal of the origin from the interference function before convolution is equivalent to scaling down the data set of structure factors. The identity expressed in (2) and (3) is not altered,

$$\mathbf{G}\gamma^*(\mathbf{F} + \mathbf{C}) = (1 - \gamma)(\mathbf{F} + \mathbf{C}) \quad (5)$$

The source of the dependence of an individual convoluted structure factor on the original unconvoluted one was identified in (4). However, the use of the γ-corrected interference function $\mathbf{G}\gamma$ instead of $\mathbf{G}$, ensures elimination of the ghosts of the original structure factors. It thus establishes the independence between corresponding structure factors before and after the convolution. Usage of $\mathbf{G}\gamma$ instead of $\mathbf{G}$ therefore allows the σ_A weighting scheme.

In the case of solvent flattening γ is the mean of the shape function $\mathbf{g}$ and is therefore equal to the fractional volume of the unit cell occupied by protein

$$\gamma = V_p/V. \quad (6)$$

V is the volume of the unit cell, and V_p is the total volume occupied by protein.

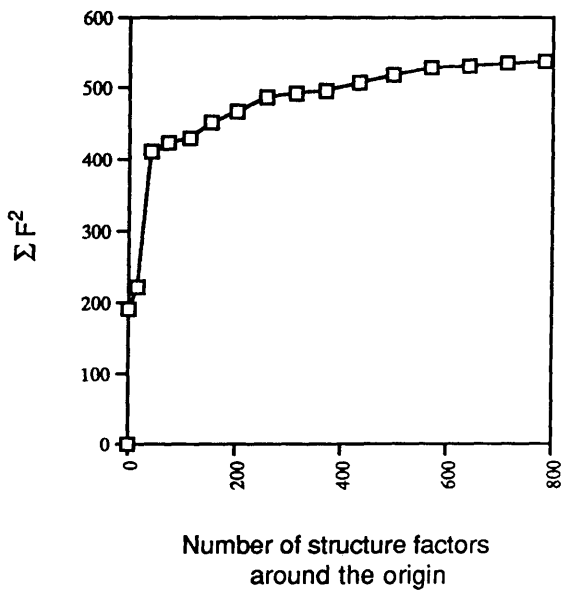


Fig. 2. Radial distribution of the accumulated power spectrum of the interference function derived from a 3.2 Å envelope of the structure of F_1 ATPase (Abrahams, Lutter, Leslie & Walker, 1994). On the horizontal axis the number of structure factors, in spherical bins around the origin is denoted, the vertical axis shows the integral of the squared structure-factor amplitudes within these bins on a linear, arbitrary scale.

Setting the origin of a symmetrical function in reciprocal space to zero is equivalent to subtracting the mean from all grid points in real space. For example, in the case of a crystal containing 50% solvent, half of the grid points of $\mathbf{g}$ is set to '1', identifying the shape of the protein, whilst the remainder is set to '0'. After subtraction of the mean of $\mathbf{g}$, which in this case is '0.5', half of the grid points will be set to '0.5', whilst the other half will be set to '–0.5'. Upon re-scaling (which does not essentially alter the outcome), the grid points defining the protein will be set to '1', and those defining the solvent will be set to '–1'. Multiplication of this $\mathbf{g}\gamma$ function with the density map therefore results in the inversion of solvent features, whilst the protein fraction will remain unchanged. See also Fig. 3. Therefore, in the case of solvent flattening, the usage of $\mathbf{G}\gamma$ instead of $\mathbf{G}$ is equivalent to flipping features in the solvent region. The solvent flipping factor k_{flip} as defined in Abrahams & Leslie (1996) can be calculated from γ,

$$k_{\text{flip}} = \gamma/(\gamma - 1). \quad (7)$$

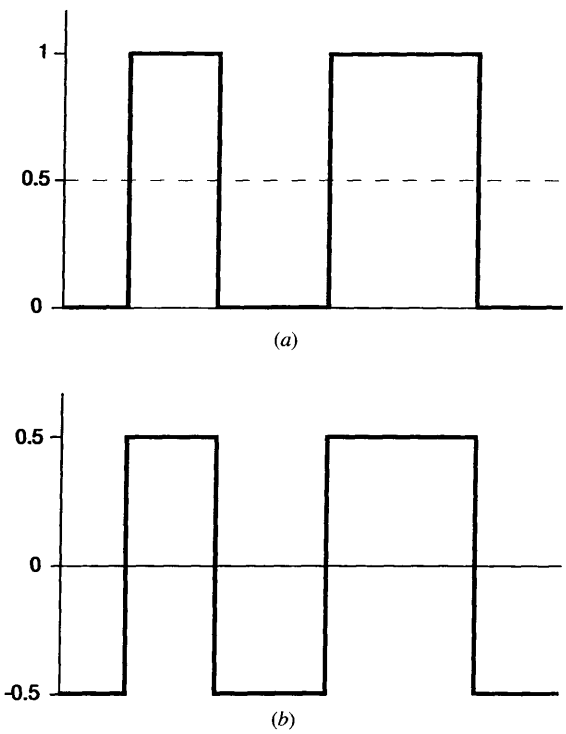


Fig. 3. A shape function $\mathbf{g}$ that defines the protein and solvent regions takes on the values '0' and '1', and has a mean value denoted by 'γ'. After subtracting γ from $\mathbf{g}$, grid points defining solvent will take on negative values, whilst those defining the protein region will be positive. Upon multiplication of the g corrected function $\mathbf{g}\gamma$ with the density function $\mathbf{f}$, features in the solvent region will be flipped with respect to the protein region. A one-dimensional example is given, with the uncorrected function in (*a*), and the γ-corrected function in (*b*).

The predicted variation in k_{flip} with solvent content closely matches the optimum refined values of k_{flip} in a range of crystals with solvent contents between 30 and 70%, judged from the experience of many users of the solvent-flipping procedure.

The function **G*****C** in (3) defines the mean difference between the protein and solvent density. Since **C** is only non-zero at the origin, and since **G** does not extend far into reciprocal space, only low resolution terms are significantly affected by this additive term.

4. Gγ and non-crystallographic symmetry

The operator describing non-crystallographic symmetry in real space also defines relationships between structure factors in reciprocal space (Bricogne, 1974). However, these related structure factors represent independent measurements. Averaging therefore reduces the bias component of the original structure factor independently from the mixing of structure factors by solvent flattening. This results in a correction of γ,

$$\gamma = \gamma_{prot}(1/n), \tag{8}$$

γ_{prot} is the mean value of the shape function of all protein in the asymmetric unit and is independent of the non-crystallographic symmetry, the factor $(1/n)$ defines the volume fraction of unique density in the protein region. If, for example, a protein complex contains six identical subunits related by non-crystallographic symmetry, and one unrelated subunit, the factor $(1/n)$ will represent the fraction relative to the volume of all seven subunits, of the volumes of only one of the symmetrical subunits, together with that of the unrelated monomer.

Sometimes it can be beneficial to use non-crystallographic symmetry to restrain, rather than to constrain a structure. For example, one can locally weight averaging by the likelihood that related electron density is identical. See for example Abrahams & Leslie (1996), where this procedure was implemented as part of a density modification protocol, or Abrahams (1996*a*), where it was implemented in atomic refinement. Here, the fraction of unique density in the protein region is larger than the inverse number of monomers, implying that γ should be decreased less than in the case of un-weighted averaging. Precisely this behaviour can be observed in Figs. 3(*c*) and 3(*d*) of Abrahams & Leslie (1996).

5. Gγ and histogram matching

The density-modification technique of histogram matching sharpens a map by enhancing its high-resolution components (Zhang & Main, 1990*b*). A histogram of the density is compared to that of an ideal map at the same resolution, after which the map is re-scaled in density bins to reflect the ideal histogram. Histogram matching can be implemented analogously to solvent flattening, provided that the shape function **g** in (2) is assumed not only to define the flat solvent, but also the re-scaling of protein density. If the histogram of the experimental map does not match the theoretical histogram, **g** is an analogue function; otherwise it is binary. The additive component **c** in the left hand side of (2) will, in most cases, no longer be constant over its entire volume, but take on a different value inside the protein region.

Irrespective of whether **g** is binary, the relationship defined in (4) still holds, since it only assumes that **g** is real. A map modified by histogram matching will therefore still contain a residual bias component, and this can be removed in the same way as in solvent flattening. Now however, γ has to be calculated explicitly from the shape function **g**, rather than to be inferred from the solvent content. The maps before and after density modification determine the shape function,

$$\mathbf{g} = (\mathbf{f}_m + \mathbf{c})/(\mathbf{f}_o + \mathbf{c}). \tag{9}$$

Here, $\mathbf{f}_o$ is the original density before histogram matching and solvent flattening, and $\mathbf{f}_m$ is the modified density. The function **c** needs to be adjusted in the protein region so that at any grid point where $(\mathbf{f}_o + \mathbf{c})$ is zero, the sum $(\mathbf{f}_m + \mathbf{c})$ is zero too.

Since a histogram provides independent constraints, γ will be reduced if this information is used together with constraints on solvent flatness.

6. Gγ and Sayre's equation

When data to atomic resolution is available (2.5 Å or better), constraints based on the atomic shape as defined by Sayre's equation become useful in phase refinement (Zhang & Main, 1990*a*). The same considerations as in histogram matching pertain to the residual bias component after applying these constraints and γ should therefore be calculated from the shape function **g**, which in its turn is determined as in (9).

7. Gγ and protein truncation

In protein truncation, grid points within the protein region with a density lower than a certain value are set to this value (Schevitz, Podjarny, Zwick, Hughes & Sigler, 1981). This improves phases in conjunction with solvent flattening and flipping (Abrahams & Leslie, 1996). As in solvent flattening, a certain fraction of the map is set to a constant value. The correction of γ can assumed to be additive,

$$\gamma = \gamma_p + \gamma_t. \tag{10}$$

Where γ_p and γ_t are the fractional volumes of the protein and the truncated density, respectively.

The crude technique of protein truncation is equivalent to over-shifting a more realistic modification function that sharpens electron dense areas in the protein region as in histogram matching, albeit in just two density bins. Since the γ correction also over-shifts density modification, and, in reciprocal space, structure factor changes, it probably does not make sense to apply it to protein truncation, as this would result in over-shifting twice. However, the removal of negative density by protein truncation after adding in *F*000 might prove to be an exception.

8. Gγ and atomic refinement

It is beyond the scope of this paper to discuss the shape function **g** that defines the constraints imposed by an atomic model. One of the difficulties is that the function **c** in (2) is no longer constant in the protein region in the case of phase errors. However, in many cases it might not be necessary to determine γ from **g**, and instead one can derive the residual bias component of the model to a good approximation from a comparison between the refinement *R* factor R_f and the free *R* factor $R_{\rm free}$, using the following equation,

$$R_f\{|\mathbf{F_o}|;\ |\mathbf{F_c}| - \gamma|\mathbf{F_o}|\} = R_{\rm free}\ \{|\mathbf{F_o}|;\ |\mathbf{F_c}|\} \qquad (11)$$

where $R_f\{\mathbf{X};\mathbf{Y}\} = \sum_{hkl}|\mathbf{X} - k\mathbf{Y}|/\sum_{hkl}|\mathbf{X}|$; k is a scale factor identical for Rf and $R_{\rm free}$.

Once γ is determined from (11), the model structure factors $\mathbf{F_c}$ can be corrected by subtraction of $\gamma\mathbf{F_o}$ (γ-scaled observed structure factors) before σ_A weighting. If γ were to be derived directly from **g**, there would be no need for a free *R*-factor determination.

9. Conclusions

The γ correction complements the σ_A weighting scheme by removing the model-independent correlation between $\mathbf{F_c}$ and $\mathbf{F_o}$. One of the premises of the σ_A weighting scheme is that corresponding structure factors are independent, a supposition that is usually incorrect. However, the γ correction to $\mathbf{F_c}$ establishes the required independence. Maps with a minimum bias require both the γ correction and σ_A weighting of the structure factors.

In some ways, applying the γ correction is comparable to calculating a free *R* factor: the information about an individual structure factor is not used to calculate the corresponding model structure factor, and instead one relies on the model to reconstruct this vector. However, there is an important difference: all the data are used, and though an individual structure factor does not influence its corresponding constrained model structure factor, it does contribute to the calculation of other structure factors, unlike in the calculation of a free *R* factor.

The merits of applying the γ correction have already been demonstrated in the case of solvent flattening. In this case the correction is equivalent to solvent flipping, a procedure first introduced in the program *Solomon* (Abrahams & Leslie, 1996; Abrahams, 1996*b*). However, the considerations which explain the success of solvent flipping also pertain to other techniques of refinement. In the cases of non-crystallographic symmetry, histogram matching and Sayre's equation, similar corrections can be applied, and these are already, or soon will be, implemented in a new version of *Solomon*. Note that in these latter cases not only the modification of the solvent region is over-shifted by the γ correction, but that also the modification of the protein region is exaggerated.

I am very grateful to Andrew Leslie for stimulating discussions and for carefully and critically reading the manuscript.

References

Abrahams, J. P. (1996*a*). *Macromolecular Refinement, Proceedings of the CCP4 Study Weekend*, pp. 185–191. Warrington: Daresbury Laboratory.
Abrahams, J. P. (1996*b*). *Jnt CCP4 ESF-EACMB Newslett. Protein Crystallogr.* **32**, 9–14.
Abrahams, J. P. & Leslie, A. W. G. (1996). *Acta Cryst.* D**52**, 30–42
Abrahams, J. P., Lutter, R., Leslie, A. G. W. & Walker, J. E. (1994). *Nature (London)*, **370**, 621–628
Bricogne, G. (1974). *Acta Cryst.* A**30**, 395–405
Brünger, A. T. (1992). *Nature (London)*, **335**, 472–475
Cowtan, K. D. & Main, P. (1996). *Acta Cryst.* D**52**, 43–48
Crowther, R. A. (1967). *Acta Cryst.* **22**, 758–764
Grimes, J. & Stuart, D. (1994). *From First Map to Final Model, Proceedings of the CCP4 Study Weekend*, pp. 67–76. Warrington: Daresbury Laboratory.
Read, R. (1986). *Acta Cryst.* A**42**, 140–149
Rossmann, M. G. & Blow, D. M. (1962). *Acta Cryst.* **35**, 24–31
Schevitz, R. W., Podjarny, A. D., Zwick, M., Hughes, J. J. & Sigler, P. (1981). *Acta Cryst.* A**37**, 511–512
Srinivasan, R. & Ramachandran, G. N. (1965). *Acta Cryst.* **19**, 1008–1014.
Wang, B.-C. (1985). *Methods Enzymol.* **12**, 813–815.
Zhang, K. Y. & Main, P. (1990*a*). *Acta Cryst.* A**46**, 41–46.
Zhang, K. Y. & Main, P. (1990*b*). *Acta Cryst.* A**46**, 377–381