commonly placed in the model in plausible positions according to molecular geometry, but this can be misleading to people using the coordinate set. If the atoms are included in the model, the atomic displacement parameters generally become very large, and this may be an acceptable flag for dynamic disorder. The hazard with this procedure is that including these atoms in the model provides additional parameters to conceal any error signal in the data that might relate to problems elsewhere in the model.

At high resolution, it is sometimes possible to model the correlated motion of atoms in rigid groups by a single tensor that describes translation, libration and screw. This is rarely done for macromolecules at present, but may be an extremely accurate way to model the behaviour of the molecules. The recent development of efficient anisotropic refinement methods for macromolecules by Murshudov *et al.* (1999) will undoubtedly produce a great deal more information about the modelling of dynamic disorder and anisotropy in macromolecular structures.

Macromolecular crystals contain between 30 and 70% solvent, mostly amorphous. The diffraction is not accurately modelled unless this solvent is included (Tronrud, 1997). The bulk solvent is generally modelled as a continuum of electron density with a high atomic displacement parameter. The high displacement parameter blurs the edges, so that the contribution of the bulk solvent to the scattering is primarily at low resolution. Nevertheless, it is important to include this in the model for two reasons. First, unless the bulk solvent is modelled, the low-resolution structure factors cannot be used in the refinement. This has the unfortunate effect of rendering the refinement of *all* of the atomic displacement parameters ill-determined. Second, omission or inaccurate phasing of the low-resolution reflections tends to produce long-wavelength variations in the electron-density maps, rendering them more difficult to interpret. In some regions, the maps can become overconnected, and in others they can become fragmented.

18.1.8. Optimization methods

Optimization methods for small molecules are straightforward, but macromolecules present special problems due to their sheer size. The large number of parameters vastly increases the volume of the parameter space that must be searched for feasible solutions and also increases the storage requirements for the optimization process. The combination of a large number of parameters and a large number of observations means that the computations at each cycle of the optimization process are expensive.

Optimization methods can be roughly classified according to the order of derivative information used in the algorithm. Methods that use no derivatives find an optimum through a search strategy; examples are Monte Carlo methods and some forms of simulated annealing. First-order methods compute gradients, and hence can always move in a direction that should reduce the objective function. Second-order methods compute curvature, which allows them to predict not only which direction will reduce the objective function, but how that direction will change as the optimization proceeds. The zero-order methods are generally very slow in high-dimensional spaces because the volume that must be searched becomes huge. First-order methods can be fast and compact, but cannot determine whether or not the solution is a true minimum. Second-order methods can detect null subspaces and singularities in the solution, but the computational cost grows as the cube of the number of parameters (or worse), and the storage requirements grow as the square of the number of parameters – undesirable properties where the number of parameters is of the order of 10^4.

Historically, the most successful optimization methods for macromolecular structures have been first-order methods. This is beginning to change as multi-gigabyte memories are becoming more common on computers and processor speeds are in the gigahertz range. At this time, there are no widely used refinement programs that run effectively on multiprocessor systems, although there are no theoretical barriers to writing such a program.

18.1.8.1. *Solving the refinement equations*

Methods for solving the refinement equations are described in *IT* C Chapters 8.1 to 8.5 and in many texts. Prince (1994) provides an excellent starting point. There are two commonly used approaches to finding the set of parameters that minimizes equation (18.1.4.1). The first is to treat each observation separately and rewrite each term of (18.1.4.1) as

$$w_i[y_i - f_i(\mathbf{x})] = w_i \sum_{j=1}^{N} \left(\frac{\partial f_i(\mathbf{x})}{\partial x_j}\right)(x_j^0 - x_j), \qquad (18.1.8.1)$$

where the summation is over the N parameters of the model. This is simply the first-order expansion of $f_i(\mathbf{x})$ and expresses the hypothesis that the calculated values should match the observed values. The system of simultaneous *observational equations* can be solved for the parameter shifts provided that there are at least as many observations as there are parameters to be determined. When the number of observational equations exceeds the number of parameters, the least-squares solution is that which minimizes (18.1.4.1). This is the method generally used for refining small-molecule crystal structures, and increasingly for macromolecular structures at atomic resolution.

18.1.8.2. *Normal equations*

In matrix form, the observational equations are written as

$$\mathbf{A}\Delta = \mathbf{r},$$

where $\mathbf{A}$ is the M by N matrix of derivatives, Δ is the parameter shifts and $\mathbf{r}$ is the vector of residuals given on the left-hand sides of equation (18.1.8.1). The *normal equations* are formed by multiplying both sides of the equation by $\mathbf{A}^T$. This produces an N by N square system, the solution to which is the desired least-squares solution for the parameter shifts.

$$\mathbf{A}^T\mathbf{A}\Delta = \mathbf{A}^T\mathbf{r} \text{ or } \mathbf{M}\Delta = \mathbf{b},$$

$$m_{ij} = \sum_{k=1}^{M} w_k \left(\frac{\partial f_k(\mathbf{x})}{\partial x_i}\right)\left(\frac{\partial f_k(\mathbf{x})}{\partial x_j}\right),$$

$$b_i = \sum_{k=1}^{M} w_k[y_k - f_k(\mathbf{x})]\left(\frac{\partial f_k(\mathbf{x})}{\partial x_i}\right).$$

Similar equations are obtained by expanding (18.1.4.1) as a second-order Taylor series about the minimum $\mathbf{x}_0$ and differentiating.

$$\Phi(\mathbf{x}-\mathbf{x}_0) \approx \Phi(\mathbf{x}_0) + \left\langle \left(\frac{\partial\Phi}{\partial x_i}\right)_{\mathbf{x}_0} \middle| (\mathbf{x}-\mathbf{x}_0) \right\rangle + \frac{1}{2}\left\langle (\mathbf{x}-\mathbf{x}_0) \middle| \left(\frac{\partial^2\Phi}{\partial x_i \partial x_j}\right)_{\mathbf{x}_0} \middle| (\mathbf{x}-\mathbf{x}_0) \right\rangle,$$

$$\left|\left(\frac{\partial\Phi}{\partial\mathbf{x}}\right)\right\rangle \approx \left|\left(\frac{\partial^2\Phi}{\partial x_i \partial x_j}\right)_{\mathbf{x}_0}\right| (\mathbf{x}-\mathbf{x}_0)\right\rangle.$$

The second-order approximation is equivalent to assuming that the matrix of second derivatives does not change and hence can be computed at $\mathbf{x}$ instead of at $\mathbf{x}_0$.

18.1.8.3. *Choice of optimization method*

First-order methods are generally the most economical for macromolecular problems. The most general approach is to treat the problem as a non-linear optimization problem from the beginning. This strategy is used by *TNT* (Tronrud *et al.*, 1987; Tronrud, 1997) and by *X-PLOR* (Kuriyan *et al.*, 1989), although by very different methods.

TNT uses a preconditioned conjugate gradient procedure (Tronrud, 1992), where the preconditioning function is the second derivatives of the objective function with respect to each parameter. In other words, at each step the parameters are normalized by the curvature with respect to that parameter, and a normal conjugate gradient step is taken. This has the effect that stiff parameters, which have steep derivatives, are scaled down, while soft parameters (such as *B* factors), which have small derivatives, are scaled up. This greatly increases both the rate and radius of convergence of the method.

X-PLOR (and its intellectual descendent, *CNS*) (Chapter 18.2 and Section 25.2.3) uses a simulated annealing procedure that derives sampling points by molecular dynamics. Simulated annealing is a process by which the objective function is sampled at a new point in parameter space. If the value of the objective function at the new point is less than that at the current point, the new point becomes the current point. If the value of the objective function is greater at the new point than at the current point, the Boltzmann probability $\exp(-\Delta E/kT)$ of the difference in function values ΔE is compared to a random number. If it is less than the random number, the new point is accepted as the current point; otherwise it is rejected. This process continues until a sufficiently deep minimum is found that the sampling process never leaves that region of parameter space. At this point the 'temperature' in the Boltzmann factor is reduced, which lowers the probability that the current point will move out of the region. This produces a finer search of the local region. The cooling process is continued until the solution has been restricted to a sufficiently small region. There are many variations of the strategy that affect the rate of convergence and the completeness of sampling. The primary virtue of simulated annealing is that it does not become trapped in shallow local minima. Simulated annealing can be either a zero-order method or a first-order method, depending on the strategy used to generate new sampling points. *X-PLOR* treats the fit to the diffraction data as an additional energy term, and the gradient of that 'energy' is treated as a force. This makes it a first-order method.

The first widely available macromolecular refinement program, *PROLSQ* (Konnert, 1976), uses an approximation to the second-order problem in which the matrix is greatly simplified. The parameters for each atom are treated as a small block on the diagonal of the matrix, and the off-diagonal blocks for pairs of atoms related by geometric constraints are also filled in. The sparse set of linear equations is then solved by an adaptation of the method of conjugate gradients.

The most comprehensive refinement program available for macromolecules is the same as the most comprehensive program available for small molecules – *SHELXL*98 (Sheldrick, 1993; see also Section 25.2.10). The primary adaptations to macromolecular problems have been the addition of conjugate gradients as an optimization method for cases in which the full matrix will not fit in the available memory and facilities to process the polymeric models required for macromolecules.

18.1.8.4. *Singularity in refinement*

Unless there are more linearly independent observations than there are parameters to fit them, the system of normal equations has no solution. The inverse of the matrix does not exist. Second-order methods fail in these circumstances by doing the matrix equivalent of dividing by zero. However, the objective function is still defined and has a defined gradient at all points. First-order methods will find a point at which the gradient is close to zero, and zero-order methods will still find a minimum value for the objective function. The difficulty is that the points so found are not unique. If one computes the eigenvalues and eigenvectors of the matrix of normal equations, one will find in this case that there are some eigenvalues that are very small or zero. The eigenvectors corresponding to these eigenvalues define sets of directions in which the parameters can be moved without affecting the value of the objective function. This region of the parameter space simply cannot be determined by the available data. The only recourses are to modify the model so that it has fewer parameters, add additional restraints to the problem, or collect more data. The real hazard with this situation is that the commonly used refinement methods do not detect the problem. Careful use of cross validation and keeping careful count of the parameters are the only remedy.

18.1.9. Evaluation of the model

Macromolecular model refinement is a cyclic process. No presently known refinement algorithm can remove all the errors of chain tracing, conformation, or misinterpretation of electron density. Other methods must be interspersed with refinement to help remove model errors. These errors are detected by basic sanity checks and the use of common sense about the model. This topic is discussed comprehensively in Part 21 and in Kleywegt (2000).

18.1.9.1. *Examination of outliers in the model*

Refinement-program output listings will normally provide some information on atoms that are showing non-standard bond lengths, bond angles or *B* factors. In addition, there is other software which can help identify non-standard or unusual geometry, such as *PROCHECK* (Laskowski *et al.*, 1993) and *WHAT IF* (Vriend, 1990). These are very useful in identifying questionable regions of structure but should not be completely relied on to identify errors or how the molecular models may be improved. Overall, the constraints in the model must be satisfied exactly, and the restraints should have a statistically reasonable distribution of deviations from the ideal values.

18.1.9.2. *Examination of model electron density*

Refinement of the model to improve the agreement between the observed and calculated diffraction data and the associated calculated phases should result in improved electron-density and ΔF maps. Unexplained features in the electron-density map or difference map are a clear indication that the model is not yet complete or accurate. Careful examination of the Fourier maps is essential. Interactive graphics programs such as *XtalView* (McRee, 1993) and *O* contain a number of analysis tools to aid in the identification of errors in the models.

There are several different types of Fourier maps that can be useful in the correction of the models. This topic is discussed extensively in Chapter 15.2. Usual maps include F_o maps, ΔF maps and $(nF_o - mF_c)$ maps. The Fourier coefficients used to compute the maps should be weighted by estimates of the degree of bias as described in Chapter 15.2. While ΔF maps are very useful in highlighting areas in the maps that reflect the greatest difference between the F_o's and F_c's in Fourier space, they do not show the electron density of the unit cell. Positive and negative regions of a ΔF map may be the result of positional errors of an atom or group of atoms, *B*-factor errors, completely misplaced atoms or missing atoms. F_o maps show the electron density but are biased by the current model. A $(2F_o - F_c)$ map is a combination of an F_o map