Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

A number of real-space torsion-angle refinement techniques for proteins, nucleic acids, ligands and solvent

Thomas J. Oldfield

Department of Chemistry (Molecular Simulations Inc.), University of York, Heslington, York YO10 5DD, England

Correspondence e-mail: tom@ysbl.york.ac.uk

Received 3 August 2000
Accepted 9 October 2000

This paper describes the implementation of real-space torsion-angle refinement as a tool for model (re)building. The algorithmic details and parameterization for a number of different protocols are presented, as well as the handling of special conditions. Examples illustrating the use of the algorithms show that these tools provide a great advantage over traditional methods for building macromolecular structures. All these algorithms have been implemented in *QUANTA* (MSI), currently available as version *QUANTA*98.

1. Introduction

Advances in macromolecular X-ray crystallographic techniques (MX) mean that it is now a powerful tool for biological and medical investigation. At the same time, less experienced crystallographers are using this technique to answer questions in their area of interest, requiring crystallographic methods to be made accessible to the non-expert. The advent of structural genomics (SG) projects will require almost full automation of MX techniques to handle the large number of structures. For example, the baker's yeast genome, *Saccharomyces* (Goffeau *et al.*, 1996), contains approximately 6000 open reading frames (ORFs), of which the majority are probably valid protein structures. In a more extreme example, current estimates from the human genome project suggest that there are between 25 000 and 150 000 genes. Manual elucidation of protein structures encoded by these is not feasible.

There have been considerable advances and efforts toward automation of X-ray structure solution in recent years. For example, heavy-atom solutions are near-automated with the programs *SOLVE* (Terwilliger & Berendzen, 1999), *Shake and Bake* (Weeks *et al.*, 1994), *SHELXL* (Sheldrick & Schneider, 1997) and *ACORN* (Yao *et al.*, to be published). Molecular-replacement searches are becoming routine in many cases, with many excellent packages available such as *MOLREP* (Vagin & Telplyakov, 1997) and *AMoRe* (Navaza & Saludjian, 1997). In the case of map interpretation at resolutions better than 2 Å, there is the method of *ARP*/*wARP* (Perrakis *et al.*, 1999). Oldfield (1996) has described methods for auto-tracing structures into maps of resolution 3 Å and better. Model building is becoming faster as more convenient and automated methods are developed and implemented.

Unfortunately, at the moment full automation of structure solution in the general case is not possible. There are still a number of 'interesting' proteins and protein complexes such as the ribosome that do not diffract well enough to allow the use of entirely automated techniques. Data-collection techniques are improving, particularly at synchrotron sites, and the complexity of tractable structure-determination problems is

increasing. Provision of automation will have two benefits: (i) current and difficult structures can be solved faster and (ii) these can serve as bases or building blocks for future full automation in more general cases. One should bear in mind that even if full automation were possible for today's problems, this would result in even more challenging structures being attempted.

Model-building techniques utilize functions of the form $f(\rho_{\rm obs}, \rho_{\rm calc}$, 'information'), where $\rho_{\rm obs}$ is dependent on 'observed' structure factors and phases, $\rho_{\rm calc}$ depends on atomic parameters and 'information' is the current knowledge available about chemistry and structure information as well as that experience provided by the investigator. Reciprocal-space refinement uses a function of the form $f(F_{\rm obs}$, plus other experimental data, $F_{\rm calc}$, 'information'). The information used for model building and reciprocal-space refinement can be different, but it is intuitively clear that these two approaches to optimizing the fit of the model parameters to the experiment are similar.

Let us consider the specific form of functions used for reciprocal- and real-space refinements. Most real-space refinement and model-building software uses a function of the form (LSQ function)

$$\sum(\rho_{\rm obs} - \rho_{\rm calc})^2, \qquad (1)$$

where summation is over the whole unit cell, $\rho_{\rm obs}$ is the 'observed' and $\rho_{\rm calc}$ the calculated electron density.

If $\rho_{\rm obs}$ has the form '$nF_o - (n-1)F_c$' then (1) could be written using a Fourier transformation (FT) and Parseval's theorem (Diamond, 1971),

$$\begin{aligned}\sum &\left(\mathrm{FT}\{[nF_o - (n-1)F_c]\exp(i\varphi)\} - \rho_{\rm calc}\right)^2 \\ &= \sum\left(\mathrm{FT}\{[nF_o - (n-1)F_c]\exp(i\varphi)\} - \mathrm{FT}(F_c\exp i\varphi)\right)^2 \\ &= \sum\{\mathrm{FT}[(nF_o - nF_c)\exp(i\varphi)]\}^2 \\ &\propto \sum|nF_o - nF_c|^2, \end{aligned} \qquad (2)$$

where FT is the Fourier transform of, n is an integer constant, F_o are the observed structure factors, F_c are the calculated structure factors and φ is the phase angle.

It should be noted that for model building where the electron density is not updated by refinement then this is only true for the first cycle of refinement. (2) shows that if all reflections are used for map generation and therefore in model building then this will 'corrupt' any free R test set. However, if some reflections are excluded then the resultant noise may prevent the derivation of optimal parameters. Reflections may be missing owing to series termination (*i.e.* resolution cutoff), because they were unrecorded or because they are reserved for free R-value calculation. Murshudov *et al.* (1997) suggests a compromise solution for this by 'restoring' absent reflections as dF_c, where d reflects the error in the model. This should be used with care however, as the resultant maps will suffer from some model bias.

A similar line of arguments could be used to show that maximum-likelihood refinement (Lunin & Urzhumtsev, 1984; Read, 1986; Bricogne, 1988; Read, 1990; Carter *et al.*, 1990; Murshudov *et al.*, 1997) is similar to some extent to that of real-space refinement (3). This is based on statistical theory (Stuart & Ord, 1991). The application of the Rice distribution (Rice, 1954) provides a valid description of the unknown phase error m, which is generally estimated from the free R set of experimental data,

$$\Gamma[(mF_o - dF_c) - F_c]^2 = \Gamma[2mF_o - (d+1)F_c]^2. \qquad (3)$$

Somewhat different arguments were used by Cochran (1948) to show that least-squares refinement (Hughes, 1941) and the corrections made using difference maps (Booth, 1946) are similar. Hughes's approach (least-squares refinement) can be thought as the predecessor of modern reciprocal-space optimization and Booth's approach reminds us of modern real-space refinement and model-building techniques.

Thus, there is a similarity, but not an equivalence, between real-space refinement and reciprocal-space refinement. Their advantages and disadvantages are set out below.

Real-space advantages.

(i) The technique for map generation is not important. It could be any map; for example, after density modification (*DM*; Cowtan, 1994). In this sense, recycling of model building and density modification would mean that *DM* is used as part of model building. The map could even result from *ab initio* or other theoretical calculations.

(ii) Small parts of structure can be fitted very fast and efficiently.

(iii) It is usually interactive, so radius of convergence is virtually 'infinity' providing that electron density is good.

(iv) One can add any other information easily into the refinement. The knowledge of a competent crystallographer for example is difficult to quantify and cannot be added in an algorithmic form but is implicitly exploited by an interactive program.

Real-space disadvantages.

(i) Result of model building on other parts of electron density is not immediate so sometimes it could be hard to decide if the rebuilt part of the structure is correct.

(ii) Requires time of the investigator.

(iii) Changes in the local region do not effect the map globally.

(iv) The actions of an inexperienced crystallographer can detract from the use of a semi-automated model building procedure.

Reciprocal-space refinement advantages.

(i) The whole structure can be refined simultaneously and hence the effect of a correctly fitted part of the structure on other parts will be instantaneous.

(ii) It is CPU-based and does not require investigator time.

Reciprocal-space refinement disadvantages.

(i) It is effectively a 'black-box' technique; the investigator has to rely on the quality of the program decision-making that is not necessarily optimal.

(ii) There may be a lack of feedback on problems within the refinement.

(iii) It is often not obvious how best to re-parameterize the problem and current software is neither as informative nor as flexible as it might be.

1.1. Parameterization

Model building can be considered as a problem of the optimization of a multidimensional function. In general, any optimization procedure faces two problems. Firstly, what parameters should be used to describe the system? In choosing these it is important to consider (i) that parameters should describe the system as completely as possible, (ii) they should have physical meaning and, if possible, (iii) they should be a minimum set describing the system under consideration. Secondly, what minimization method should be used to optimize the parameters? This should be practical, stable, fast and have a large radius of convergence. Often it is hard to fulfil all these requirements. For various local minimization methods, see Fletcher (1987) and references therein. For global optimization methods, see Torn & Zilinskas (1987) and references therein.

To date, refinement of macromolecular structures is generally carried out in reciprocal space by modifying the atomic parameters x, y, z and an isotropic thermal factor B to improve the agreement between the observed diffraction intensities and the estimates calculated from the atomic model. Crystals of macromolecules often do not diffract sufficiently well to provide enough observations to fit four parameters for each atom. It is usually necessary to include geometric restraints (bonds, angles, non-bonding, planarity, chiral and prochiral centres) as 'data' to improve the ratio of observations to parameters. Within the application of maximum likelihood these restraints can be considered part of the prior knowledge and are therefore incorporated into the formulation as such.

A second method of parameterizing macromolecular structures is to use torsion angles. Torsion-angle refinement requires that a root point be defined for the torsion-angle definitions (Fig. 1). Hence, all torsion angles defined from this single point are highly correlated and dependent on this root-point position. If this correlation can be overcome, the torsion-angle method of parameterization can be very advantageous, because the number of torsion-angle parameters including the root-point position and orientation is generally much smaller than the equivalent number of xyz parameters required for the same system. In essence, the covalent structure is constrained within torsion-angle refinement.

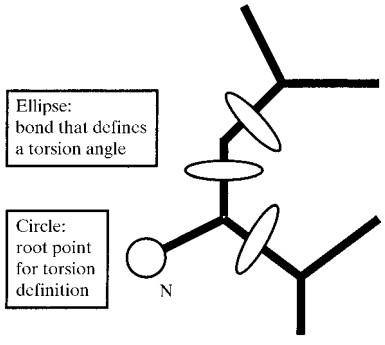

Figure 1
Torsion-angle definitions for the amino acid leucine with the root point defined at the main-chain N atom.

Any change of the xyz parameters will obviously change the stereochemistry of a molecule, *i.e.* the bonds, angles, planes and chiral centres. When the ratio of parameters to observations is low, it is necessary that these stereochemical terms be restrained during the refinement process. Therefore, xyz refinement is a compromise between correction of geometry and fitting the experimental data. On the other hand, when the model is made up of constrained rigid units, there is no correlation between these and the value of the torsion angles. Since nearly all the flexibility within proteins results from the variability about rotatable bonds, separating the geometric terms completely from the refined parameters has obvious advantages. It should also be recognized that although most deviations within a protein structure can be described by rotation of torsion angles, there is still some flexibility within the covalent subunits of proteins. Refinement that does not allow any variation in the constrained building blocks will probably fall into false minima at some point. The number of torsion angles within a protein is relatively small (compared with the number of xyz parameters), leading to a high over-determinacy of the refinement problem. There is also the problem of the computational difficulty of generating derivatives of the torsion angles within non-orthogonal coordinate systems, *i.e.* the map. Although this is only a technical problem that affects the calculation time, since model building is supposed to be interactive, it is actually a critical problem here.

1.2. Refinement programs

Reciprocal-space least-squares $xyz(B)$ gradient-refinement algorithms incorporating geometric restraints have been implemented by several authors, the best known or these being *PROLSQ* (Konnert & Hendrickson, 1980), *X-PLOR* (3.4 and lower versions; Brünger, 1992) and *TNT* (Tronrud *et al.*, 1987). More recently, programs using a maximum-likelihood refinement target have been developed and implemented within the programs *BUSTER* (Bricogne, 1993) and *REFMAC* (Murshudov *et al.*, 1997), while the implementation of Pannu & Read (1996) has been added to *CNX* (MSI) and *CNS* (Adams *et al.*, 1997). All these programs refine parameters in reciprocal space and using $xyz(B)$ parameterization and first-derivative gradients or pseudo second-order derivatives. Real-space torsion-angle refinement was first implemented by Diamond (1971). The torsional-angle refinement method was added to *X-PLOR* 3.851 (Rice & Brünger, 1994) and the current version of this program is available as *CNX* (MSI). More recently, real-space targets using a molecular-dynamics search protocol (Chen *et al.*, 1999) has been implemented in *X-PLOR* by Chapman as a method of refining protein structures.

1.3. Model-building programs

To date, model-building programs have been mostly devoid of useful automated techniques and so manual model building represents the human decision of the required shifts necessary to remove the model from local minima. Examples include *FRODO* (Jones, 1985), *O* (Jones & Kjeldgaard, 1997), *CHAIN* (Sack & Quiocho, 1997), *XtalView* (McRee, 1999), *MAIN* (Turk, 1992) and *QUANTA*96 (Oldfield, 1996). A method of real-space torsion-angle refinement has been implemented in *O* (Jones & Kjeldgaard, 1997) using a grid summation (Jones *et al.*, 1991) to score the quality of fit for different rotamers of amino-acid side chains as well as more recent improvements. McRee has implemented both a C^{α} refinement method for tracing as well as a general real-space refinement technique within the program *XtalView*. This algorithm uses *xyz* refinement against electron density with geometrical restraints. Refinement of geometry as a modelling tool has been implemented and the best known of these is *Sculpt* (Surles, 1994), although an implementation is available in *Xtalview* and to a limited degree in other programs.

It is sensible to include automated real-space refinement during the manual model-building process to provide a rapid and less subjective means of placing atoms. The technique should not detract from the process of reciprocal-space refinement by doing the same work in real space. The overall refinement process is much more efficient if different parameterization is used for the real-space and reciprocal-space refinement. This is performed by parameterizing the real-space refinement using torsion angles, whereas the reciprocal-space refinement is parameterized in $xyz(B)$.

This paper will concentrate on fast, efficient and automatic real-space refinement techniques used for model (re)building as well as *de novo* map tracing. The methods are all based on torsion-angle parameterization and allow flexibility within the covalent subunits of structure as well as removal of correlated affects associated with polymer structures. The torsion-angle methods allow the automated methods of fitting protein, nucleic acids, ligands, solvent, C^{α} traces and secondary-structure element refinement into electron-density maps.

2. Methods

The real-space refinement approach described in this paper is applied in an *ad hoc* but effective way. Any map of the form $nF_o - (n-1)F_c$ or any σ_A-weighted map will show a gradient in the neighbourhood of incorrectly placed atoms. By improving the fit between the calculated atom density and such a map it is possible to both correct the model and to allow the user to visualize what is being performed. It has the advantage that sections of the model can be refined independently of others and the underlying assumptions of sequence and geometry can be modified during the refinement. Full descriptions of the subunits for proteins, nucleic acids and common ligands are available within the program. For other classes of ligands such as sugars there are three other methods of parameter selection.

2.1. Parameters

The refinement techniques described use a mixed parameterization method to solve a number of problems associated with both $xyz(B)$ and torsional angle parameterization of macromolecular structure. The refinement protocol carries out interspersed cycles of geometry refinement using restraints on the stereochemistry followed by constrained torsion-angle refinement of each separate amino acid (or nucleic acid). These interspersed cycles of refinement are entirely independent and therefore require no weighting scheme. During the amino-acid (or nucleic acid) torsion-angle refinement the additional degrees of freedom associated with the rigid-body movement and rotation of each amino acid are also refined. Since the rigid-body movement and rotation of each amino acid is defined along the principal components of the amino acid, these six degrees of freedom are redefined at each step. Although this appears to be inefficient, requiring the parameterization of both the geometrical terms and rotatable bonds, it does allow large movements of the residues during the refinement to experimental data while retaining the integrity of the polymer structure. The inclusion of cycles of refinement where geometry is restrained also allows flexibility within the covalent structure of the macromolecule. This is important during the initial stages of modelling, where significant errors in the structure cannot be corrected by torsion-angle shifts alone.

The ability to handle data in a transparent way without the crystallographer having to set up any parameters or special conditions is important in a model-building program. All models are checked at the beginning and end of the refinement methods, allowing all higher level functionality to be separated from the model structure description. All parameterization is carried out during the data-checking steps. The check-in and check-out routines not only generate the necessary standard description of monomer units for refinement to proceed, but they also detect and handle alternate conformations, missing atoms, extra atoms, protonation state, polymer patching, RNA/DNA differences and modified bonds.

2.1.1. Standard parameters. To carry out torsion-angle refinement it is necessary to give a full set of definitions of rigid units, rotatable bonds and permissible limits. A highly minimized template using the Engh & Huber (1991) parameters defines amino acids and nucleic acids. This file defines all rotatable bonds within each residue construct and hence provides a full description of all geometrical and torsional angle parameters. This allows the refinement of extremely distorted initial models where the close contacts have no relation to the required connectivity. The template files are extensible as they are free formatted.

The geometry of polar and all hydrogen-ligand models are automatically determined using a look-up table of internal coordinates (IC): bonds, angles and improper torsion definitions based on atom typing. The IC definitions can be taken from those of Engh & Huber (1991) or from the *X-PLOR*98 (MSI) definitions. The rotatable bonds are determined auto-

matically using a rule-based method of valency analysis, taking into account the type of hydrogen model in use. Since there are no suitable IC tables of no-hydrogen ligands, these molecules cannot be automatically parameterized by this method. Instead, the program assumes that the current geometry is correct when first 'touched' and writes a residue topology file (RTF) file for any subsequent refinement of this residue. The rotatable bonds in these molecules are determined using the same rule-based method of valency.

All restraints on bond deviation (4), deviation in the open angle (5) and deviation of the improper torsion angle (6) for chiral centres, prochiral centres and planar atoms are quadratic in form. (7) represents the formulation of the torsion-angle restraints with multiple minima.

$$R_{[i]\text{bonds}} = W_{[i]\text{bonds}}(b_{[i]} - b_{[i]o})^2 \quad (4)$$

$$R_{[i]\text{angle}} = W_{[i]\text{angle}}(\theta_{[i]} - \theta_{[i]o})^2 \quad (5)$$

$$R_{[i]\text{improper}} = W_{[i]\text{improper}}(\zeta_{[i]} - \zeta_{[i]o})^2 \quad (6)$$

$$R_{[i]\text{torsion}} = W_{[i]\text{torsion}}[1 + \cos(n\varphi - \delta)]^2. \quad (7)$$

$b_{[i]}$, $\theta_{[i]}$ and $\zeta_{[i]}$ are the current values for the bond length, angle and improper torsion angle, respectively, for the ith atom. $b_{[i]o}$, $\theta_{[i]o}$ and $\zeta_{[i]o}$ are the optimal values for the bond length, angle and improper torsion angle, respectively, for the ith atom. n defines the number of minima about a torsion angle, δ is the deviation in torsion angle and $W_{[i]}$ is the weight used for atom i.

Derivatives are calculated from these quadratic functions. The derivatives for non-bonding interactions are based on finite differences as this allows the determination of continuous derivatives for restraints. There are three non-bond formulations. The first is based on a Lennard–Jones 6–12 potential for van der Waals interactions plus an electrostatic term (using a radial distant variation in dielectric; equations 8 and 9), the second uses just the van der Waals term (8) and the third uses a simple repulsive formulation (10).

$$R_{[i,j]\text{van der Waals}} = W_{\text{vdw}}\varepsilon_{ij}\left[2\left(\frac{\sigma_{ij}}{r_{ij}^{12}}\right)^6 - \left(\frac{\sigma_{ij}}{r_{ij}^{6}}\right)^{12}\right] \quad (8)$$

$$R_{[i,j]\text{electrostatic}} = W_{\text{elec}}\left(\frac{q_i q_j}{4\pi\varepsilon_o\varepsilon_r r_{ij}^2}\right) \quad (9)$$

$$R_{[i,j]\text{push}} = W_{\text{push}}(r_o - r_{ij}) \text{ for } r_{ij} < r_o. \quad (10)$$

σ_{ij} is the equilibrium separation between atoms i and j. This is estimated using the combination rules: $\sigma_{ij} = (\sigma_i/2 + \sigma_j/2)$. ε_{ij} is the minimum energy at the separation $r = \sigma_{ij}$. This is estimated using the combination rule $\varepsilon_{ij} = \varepsilon_i^{1/2}\varepsilon_j^{1/2}$. q_i and q_j are the partial changes for the atoms i and j. r_{ij} is the separation between the atoms i and j. W_{vdw}, W_{elec} and W_{push} are non-standard weights used to scale the non-bond terms with respect the other energy terms to soften the interaction.

While density-fitting macromolecular structure it is best to exclude electrostatic interactions when the protein and solvent model is incomplete. The complete formulations for non-bonding are provided, but their use is not recommended for crystallographic map interpretation. The simple repulsive formulation results in fast minimization without loss of precision.

The non-bond interaction lists are determined using a lattice-search algorithm, with all previously defined 1–2, 1–3 and 1–4 interactions excluded. Since the algorithms described are designed to work with subsets of atoms, the non-bond lists are divided into an intra-non-bond list for the refined atoms and an inter-non-bond lists for interactions between the refined set and all other atomic positions. This includes those generated by symmetry and non-crystallographic symmetry (NCS).

The occupancy and B-value contribute to the refinement by weighting the fit to density for each atom. An additional weight is also provided where the resolution is low (below 3.5 Å) as a quadratic term scaling an atom's occupancy as a function of its distance from its parent C^α atom. This has no physical basis, but has been found in practice to mirror the experimental observation that side-chain density tends to be truncated at lower resolution. Without this weighting, any refinement of proteins with low-resolution data results in the requirement to fit the side chain forcing the main-chain atoms out of density.

2.1.2. Special case parameterization. The restraint setup for the peptide bond in proteins automatically detects the presence of *cis*/*trans* conformations in the model; the restraint applied to maintain partial planarity of this bond depends on the starting conformation. It is also possible to enforce either a *cis* or *trans* conformation. An improper restraint is generated to maintain the approximate planarity of the peptide group as well as the inter-peptide bonding and angle parameters.

Restraints for disulfide bonds are automatically added on detection of this type of covalent bond. It adds a single bond parameter and two angle parameters. If either of the cysteine residues, but not both, are present in a refinement zone then the program will include the other cysteine in the refinement and add the necessary restraints. As with the peptide bond (and the phosphoester bond in nucleic acids), the disulfide-bond restraint is removed during some cycles of refinement to break the correlation in the region of these restraints.

Any user-defined fixed atoms are constrained within the geometrical refinement stage and restrained using a quadratic function during the torsion-angle refinement. Fixed atoms are automatically defined where the refined zone covalently joins an unrefined zone. This not only allows the user specification of fixed points within the refinement protocol but also fixes the end points of the refinement zone within the polypeptide/polynucleotide chain.

It is possible to include restraints about the torsion angles φ and ψ in proteins (11 and 12). This allows restraints of 'helix' and 'sheet' conformations that define a single minima torsion-angle restraint about these torsion angles. These restraints allow regions of polypeptide chain to be slowly driven towards a required conformation even when there is little or no density.

$$R_{\text{Ramachandran}}(\varphi_i) = W_{\text{Ramachandran}} * (\varphi_{[i]\text{current}} - \varphi_{[i]\text{best}}) \quad (11)$$

$$R_{\text{Ramachandran}}(\psi_i) = W_{\text{Ramachandran}} * (\psi_{[i]\text{current}} - \psi_{[i]\text{best}}). \quad (12)$$

$W_{\text{Ramachandran}}$ = 0.1 kcal (1 kcal = 4.184 kJ). For allowed region restraints, $\varphi_{[i]\text{current}}$ and $\psi_{[i]\text{current}}$ are the current Ramachandran torsion angles; $\varphi_{[i]\text{best}}$ and $\psi_{[i]\text{best}}$ are the nearest point in the allowed Ramachandran region to the current point. For specific secondary-structure restraints, helix $\varphi_{[i]\text{best}} = -58^\circ$ and $\psi_{[i]\text{best}} = -48^\circ$; sheet $\varphi_{[i]\text{best}} = -115^\circ$ and $\psi_{[i]\text{best}} = 120^\circ$.

It is also possible to apply restraints that drive all φ/ψ torsion angles in a refinement zone to the nearest point on the Ramachandran surface (Ramachandran & Saisiskharan, 1969). Before each cycle of refinement, the application determines the nearest allowed region on the Ramachandran surface to the current conformation of φ and ψ. A single-minima torsional restraint is then applied for the next cycle of refinement that drives each φ and ψ towards this required value. The restraint has to be updated during each cycle as correlation may change the shortest path to the allowed region between each cycle and may change the number of φ and ψ angles that require restraint and the restraint value.

Alternate conformations are handled by splitting and creating full-residue models for each alternate form of each residue regardless of the number of atoms with different positions at the starting point. The conformation that is passed to the refinement algorithm is defined by the atom picked by the user unless this is ambiguous, in which case a prompt is issued requiring the user to make a decision. On completion of refinement, the alternate descriptions of residues are reformed, where atoms are merged if less than 0.01 Å in separation. This method will handle any number of alternate conformations for a residue and allows separate operations to be carried out for each alternate state.

The presence and absence of some atoms depends on the type of models used. For example, RNA/DNA differs by the presence/absence of an O atom on the sugar ring. Other conditions involve the termini patching of residues, plus the mode of protonation of the molecule and, in the case of nucleic acids, the presence/absence of a 3′-terminal phosphate group. It is necessary therefore to handle these as special conditions so that the user does not need to decide how to handle each individual case. All special conditions are handled so that the 'standard' model of a residue is passed to the refinement algorithm; the special residue form is then generated geometrically on 'check-in' of the data to the data structure.

Missing atoms are automatically generated. If a 'checked-out' residue contains less than the required number of atoms, (including the condition of a single C^α atom), then a function will automatically place all other atoms geometrically. The refinement will then optimize the placement of the new atoms and the new atomic coordinates are 'checked-in'. The same occurs for undefined atoms. These are atoms marked as having an 'x' coordinate greater than 998.00 in the data. In this case, the actual coordinate is ignored; instead, the initial coordinate starting point before refinement is generated by geometry. Extra atoms are removed. This technique provides a way to automatically build H atoms or remove H atoms from the model.

2.2. A density-fitting function

The application uses two forms of interpolation for the electron density. The first is a linear and the second is a quadratic interpolation. Linear interpolation tends to produce poor approximations for a final atom position as an atom will tend to move to the highest density point, but has the advantage that it results in a continuous function with no false minima, which are often found with quadratic interpolation. Hence, all refinement uses the linear interpolation until near-convergence, then switches to the more precise quadratic interpolation of the electron density and continues the refinement to tolerance completion.

Since the electron density is stored as a grid map that is not necessarily orthogonal, it is necessary to transform atoms and map to the same axial system. It would not be efficient to transform all or part of the map to an orthogonal grid, as this would be extremely computationally expensive or require large amounts of memory. Hence, the algorithm transforms the atomic positions into map space during refinement and atom-fit values are determined by interpolation. Since the torsion-angle parameters are defined in orthogonal 'world' coordinates, while values and gradients are defined from interpolated points defined from the torsion angle and thus world coordinate system, there is no xyz parameter dependence with non-orthogonal maps.

2.3. Gradient-refinement protocol

Gradient refinement is a non-linear method that iterates until a tolerance level is achieved or the cycle limit is exceeded. Hence, a gradient-refinement algorithm takes steps along the electron-density gradient to always improve the overlap of the molecule. Since the procedure is general, the gradient refinement can be used to refine any size of molecular fragment from a single water to an entire protein/polynucleotide.

The gradient-refinement protocol proceeds in cycles between free-monomer refinement to map and refinement of geometric restraints. During free-monomer refinement, each monomer unit is parameterized using three positional (centre of mass), three orientation (about principle components of monomer) and freely rotatable bonds. The geometric terms are constrained. During the geometry-refinement stage, the entire fragment to be refined is restrained using bonds, angles, torsion angles, improper angles, planes and non-bonds.

As stated before, the refinement of model to map is carried out in non-orthogonal map space and is performed by steepest decent with derivatives determined by finite differences. Refinement of the geometric terms uses the line-minimization method of Brent (1973), with bracketing adapted to use derivatives determined from the quadratic approximations for restraints. No scaling between the fit to data and geometric terms is required, as the two refinements occur independently.

Force constants for bonds, angles, impropers and torsional angle terms are taken from the Engh and Huber or CHARMM tables, while non-bond terms are softened and limited to prevent them dominating the refinement.

2.4. Grid-refinement protocol

The second protocol is grid real-space torsion-angle refinement. The principle of this type of calculation is to find the best fit to density by systematically varying the torsion-angle parameters. This method of refinement will always find the same answer regardless of the starting conformation and so in essence has an infinite radius of convergence.

In practice, as the number of torsion angles to fit increases, the number of density fits requiring evaluation quickly reaches an impossibly large number. With a precision of 2°, n torsion angles will require 180^n density-fit calculations (and non-bond calculations). The grid refinement as implemented here uses some short cuts that allow four torsion angles to be fitted (a lysine/arginine side chain) in approximately 0.3 s with a precision of 2° (R4000 Indigo, 100 MHz). Hence, it is possible to fit side-chain atoms of amino acids and nucleic acids and the peptide planes of proteins using this protocol.

2.4.1. Fitting amino-acid side chains and nucleic acid bases. All geometric terms are constrained to the initial values and are therefore unrefined. The algorithm performs a two-stage tree search progressively fitting each torsion angle. For each torsion angle fitted, two open angles are defined about the second atom that defines the rotatable four-atom set (Fig. 2).

The two opening angles are defined about (N—C^{α}—C^{β}). The first opening-angle axis is defined as orthogonal to the three atoms set and through the C^{α} atom. The second axis is orthogonal to the three-atom set and orthogonal to the first axis and through the C^{α} atom (Fig. 2). These opening angles allow a greater convergence radius where there is error in the C^{α} position or chiral volume. These errors can often prevent the placement of side-chain atoms by changing torsion angles alone. Starting with the first torsion angle in the side chain (N—C^{α}—C^{β}—?$^{\gamma}$), the torsion angle is varied with a precision of 10°. The opening angles are searched at −8°, 0 and +8° about the initial value. The occupancy of the atoms further down a side chain is set to 0. The best solution of this fit is then further refined by ±6° with a precision of 2° about the opening angles and ±8° with a precision of 2° about the torsion angle. The nominal time for the calculation of a single torsion angle is less than 0.1 s. On completion of the fit of the first torsion angle, further torsion angles further in bonded distance from the C^{α} atom are fitted in the same way.

It should be noted that some covalent structures can prevent the determination of the correct torsion-angle value by this method as the solution to the tree search is ambiguous. This includes the amino acid isoleucine and many ligands. When χ_1 of isoleucine is fitted, the atoms $C^{\gamma 1}$ and $C^{\gamma 2}$ are equivalent within the tree search. This is the case for any torsion angle followed by an atom with three or more bonded atoms that are equivalent or very nearly equivalent with regard to their interaction with X-rays. To resolve this problem the algorithm revisits a χ angle; in the isoleucine example χ_2 is fitted for both alternative χ_1 values.

The proline residue is fitted by rotating the five-member ring about C(−1)—N—C^{α}—C and the pucker torsion angle C^{α}—C^{β}—C^{δ}—C^{γ}. The latter is searched within the range ±30° (Fig. 3).

2.4.2. Fitting main-chain atoms. To fit peptide planes, an ideal *trans*-peptide plane is built and all combinations of the peptide plane rotated about the axis defined by the pseudo-bond $C^{\alpha}(0)$—$C^{\alpha}(1)$ are tried with 1° steps, plus variation of Ω values between ±12° also with 1° steps (Fig. 4). Note that the Ω angle distorts the peptide plane and requires redefinition of the peptide geometry after each step, although the C^{α}—C^{α} separation is not redefined by this fitting procedure.

2.5. Monte Carlo refinement

The Monte Carlo refinement protocol samples conformation space by setting the search-parameter torsion angles to random values and saving the best set of solutions of fit to density during the search. This refinement method is suitable for instances where a very large radius of convergence is required but systematic grid searching would take too long. This has been implemented for two different techniques for model building and refinement. The first is used to fit main-chain protein atoms in loops/termini and the second is to fit complex ligands to electron density. The fitting of ligands to electron density in an automated way is described elsewhere. Fitting of peptide main-chain atoms proceeds at approximately 2500 conformations s^{-1} for nine parameters and fitting of ligands at 900 conformations s^{-1} for nine torsion-angle

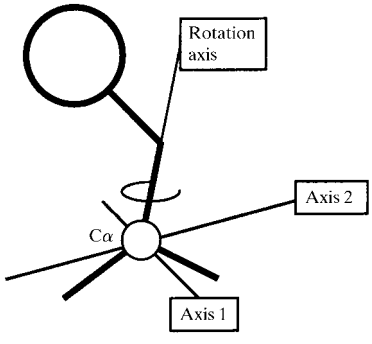

Figure 2
Torsion-axis definitions used for grid refinement of amino-acid side chains.

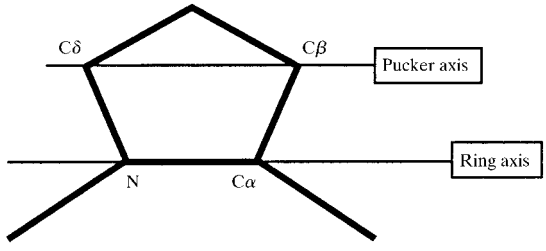

Figure 3
Torsion-axis definitions for a proline imino acid.

parameters including the optimization of position and orientation for each fit (R4000 Indigo, 100 MHz). This allows the fitting of up to eight residue loops (and termini) in proteins and ligands with up to 16 rotatable torsion angles in an acceptable time.

2.5.1. Loop fitting in proteins. Since the refinement proceeds in a driven random way, any correlation between the torsion angles does not prevent refinement but does limit the number of simultaneous searched torsion angles. The values of the bonds, angles, improper torsion angles, chiral centres and planes are set to ideal values before starting and are constrained (*i.e.* unrefined). The φ and ψ torsion angles within the search zone are defined as search parameters and Ω is constrained to 180°. The non-bond interactions to the rest of the model coordinates are required for each conformation and since they are computationally expensive these are calculated once before any searching. The non-bond interactions with the moving atoms are calculated for the fixed atoms at grid points within a suitable map volume about the starting loop conformation. That is, a non-bond energy surface is generated on the map grid and this is used while exploring conformational space of the loop under refinement to give a simultaneous energy term for non-bond clashes and map fit value.

The subtractive method for random numbers by Knuth (1981) is used to generate a satisfactory series of non-repeating random numbers. (Note that for nine parameters at 2500 conformations s^{-1} running for 1 h it is necessary that the random-number sequence does not repeat for 81 million calls.)

For each torsion-angle parameter, a random value is generated between 0 and 359° in 1° steps and the polyalanine chain conformation set to reflect these. If a loop is to be fitted, then the length of the polyalanine chain is determined and checked to be within ±1 Å of the distance between the loop − 1 and loop + 1 residues. If this length is satisfied, then the first residue is superimposed on the original conformation. (Note that if an N-terminal region is being fitted, then the last residue is fitted to the original conformation.) The next stage is to determine the intra-non-bond value using the sum of all interactions of all the atoms in the loop with themselves. This effectively determines whether the conformation has all residues lying approximately within Ramachandran-allowed regions. If this restraint is satisfied, then the fit of the random conformation to the electron density is determined in real space. Since the inter-non-bond interaction is encoded within this electron density, the function calculated is a sum of intra-non-bonds and the fit to density. At any one time, the best ten solutions are saved.

2.6. Hydrogen spin density fitting

A simple method of H-atom placement has been implemented. None of the previously described algorithms take any account of H atoms. Even if coordinates are available in the model their occupancy is set to zero, as most experimental data does not contain sufficient information to place these. The H-atom placement is separated from general structure modelling as information about hydrogen positions often results from different experiments (*i.e.* neutron diffraction) from those used to determine the rest of the atom positions. For all residues where H atoms are to be included, the density fitting algorithm carries out a 'H-atom spin' analysis. This requires that a rotatable bond can be defined where the fourth atom is a H atom. The torsion angle is rotated through a 1° search and the best fit for the H atom(s) to the density is taken as the solution.

2.7. C^{α}-atom refinement for tracing proteins

The refinement algorithms described are also used for tracing C^{α} atoms into electron density as part of the *de novo* map interpretation. When placing a C^{α} atom into electron density it is refined using a gradient algorithm. Four atoms are defined for the placement; the C^{α} atom and three tetrahedral atoms defined for an L amino acid. These are refined using the gradient about a pseudo $C^{\alpha}-C^{\alpha}$ bond on the surface of a sphere of variable radius between 3.0 and 4.3 Å in diameter (Fig. 5). All valid pathways within the electron density (in preparation) define the starting point for the refinement for a C^{α} atom in tracing.

Gradient refinement has also been implemented to refine the fit of a whole C^{α} trace into electron density. It is used to improve the fit of the C^{α} atoms while a weak 3.8 Å distance geometry restraint is used between C^{α} atoms. Convergence is improved by adding virtual atoms of 0.5 occupancy at 1/3 and 2/3 positions along a $C^{\alpha}-C^{\alpha}$ pseudo-bond to represent a peptide plane of undefined orientation. Torsion-angle parameters are defined for all four consecutive C^{α} atoms and

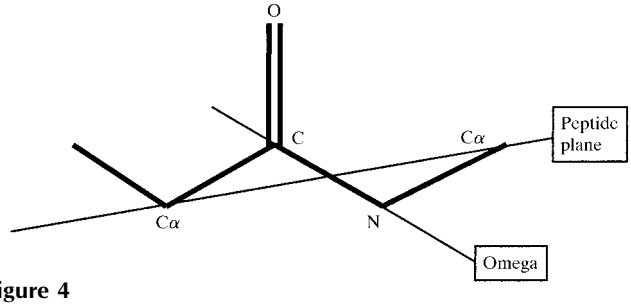

Figure 4
Torsion-axis definitions for a peptide plane.

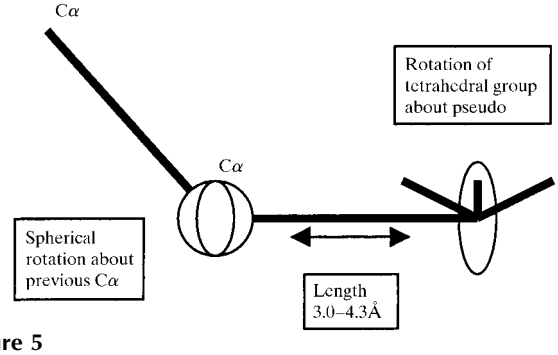

Figure 5
Gradient-refinement parameters used for C^{α} placement during C^{α}-tracing maps.

angular parameters are defined for all three consecutive C^{α} atoms as an axis perpendicular to the three defining C^{α} atoms (Fig. 6). This torsional angle and angular refinement allows the trace to rotate about bonds and flex about each C^{α} atom.

2.8. Refinement as a tool for converting C^{α} traces to all-atom models

The grid-refinement method has been implemented in such a way that it can be used to convert C^{α}-trace atoms into all-atom models. A tool automatically takes a C^{α}-trace segment, places all the peptide planes by grid refinement as described above and then adds the side-chain atoms using the grid refinement defined above. This allows the generation of all-atom models from a C^{α}-atom trace at about 10 residues s^{-1} without user intervention.

3. Application of refinement to data

The refinement methods described above are used by tools within the X-Build, X-Autofit and X-Ligand functionality of *QUANTA*. The user selects any part of the structure to refine by picking two atoms to define the start and end points of refinement. The atomic coordinates for these residues are 'checked-out' of the full data structure, multiple conformations are split up and any special conditions of the residues are recorded. Each residue of the refinement zone is checked to guarantee all atoms are present and any terminal patches removed. The zone is checked for disulfide and modified bonds and the bonded partner residues added to the refinement zone. Non-bond lists are generated for all neighbouring atoms, crystallographic symmetry and NCS and written to a linked list as a lattice. Torsion angles are defined while bond, angle, chiral centre, plane, Ω and any other restraint options are generated for the required zone. Depending on which protocol required (gradient, grid or Monte Carlo) the refinement proceeds as described.

On completion of the refinement, the terminal patches are rebuilt geometrically, alternate conformations merged, special conditions reapplied and the coordinates checked back into the full data structure.

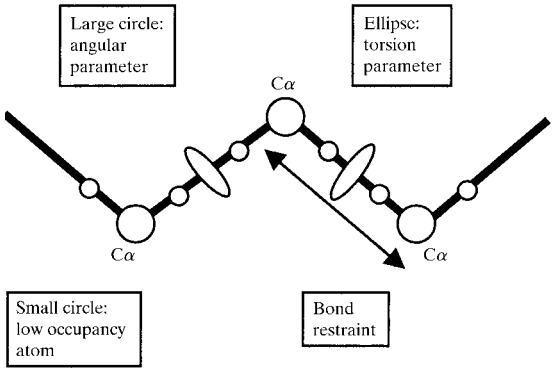

Figure 6
C^{α}-trace refinement parameterization.

3.1. Real-space grid refinement of proteins and nucleic acids

Figs. 7(*a*) and 7(*b*) show the placement of a tyrosine residue into a 2.5 Å experimental map of ribonuclease (Sevcik *et al.*, 1991) using real-space torsion-angle grid refinement. The refinement time is approximately 0.1 s and the shifts are in excess of 5 Å. It is possible to use a second tool to drag C^{α} atoms (or main-chain atoms), while placing the side-chain atoms at a rate of ten placements per second. This allows the dragging of root atoms until the C^{α} position or chiral volume is optimized so as to allow the side chain to fit. A third tool is provided to carry out real-space torsion-angle grid refinement on peptide planes.

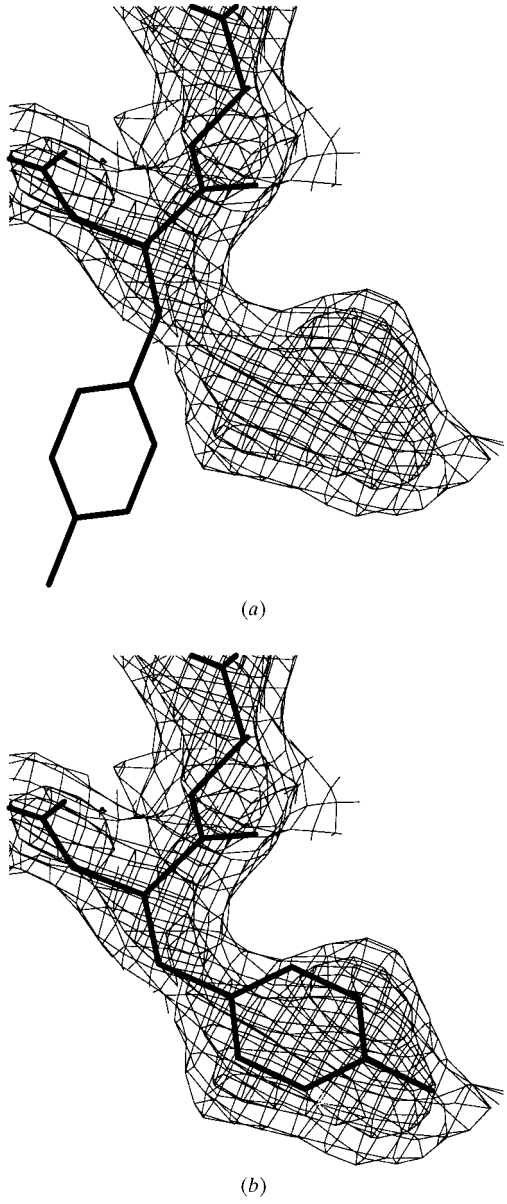

Figure 7
Real-space grid refinement as a tool to place a single tyrosine side chain into electron density. (*a*) The position of the residue before refinement; (*b*) the residue coordinates after refinement.

3.2. Real-space gradient refinement of proteins and nucleic acids

Figs. 8(*a*) and 8(*b*) show model protein coordinates before and after fitting to a 2.5 Å experimental map of ribonuclease (Sevcik *et al.*, 1991) using real-space torsion-angle gradient refinement. The protocol required the use of a single tool that carries out a gradient refinement on the four amino acids; the tool was used twice in the example shown. The refinement tool will allow shifts in atomic position of approximately 2 Å and in this case some atoms have moved over 2.5 Å during the two refinement cycles, with a total refinement time of 20 s.

3.3. Gradient refinement of ligands

Gradient refinement is applied to ligands using a single general tool for refining just one residue of any type (amino acid, NA, water or ligand). The ligand shown in Figs. 9(*a*) and 9(*b*) is that of methyl-*para*-benzene bound to insulin (Whittingham *et al.*, 1995). The refinement time is about 2 s, resulting in atomic shifts in excess of 2 Å. The radius of convergence expected during torsion-angle gradient refinement is inversely proportional to the number of correlated torsion angles to fit. In this simple case with two rotatable bonds, rotation shifts of up to 120° are possible, but with more complex ligands these become significantly limited and Monte Carlo refinement is more appropriate.

3.4. Real-space Monte Carlo refinement of proteins

The Monte Carlo refinement protocol as described has been implemented for use in the fitting of loops and termini in proteins, as well as fitting ligands to electron density by conformation searching. The fitting of ligands to electron density using Monte Carlo refinement forms an application and this procedure (X-Ligand) is presented elsewhere (in preparation). Figs. 10(*a*) and 10(*b*) show the use of Monte Carlo refinement as applied to a badly modelled loop of five amino-acid residues within a 2.5 Å experimental map of ribonuclease (Sevcik *et al.*, 1991). Monte Carlo refinement is used where the exhaustive grid search of torsion angles would represent an impossibly long calculation and where the shifts required to correct the model are far too large for gradient refinement. The protocol used to move the model coordinates shown in Fig. 10(*a*) to give those shown in Fig. 10(*b*) is as follows.

A single tool is used to carry out a Monte Carlo refinement of the five-residue loop. This was allowed to run for 5 min, resulting in the screening of just under 0.6 million conformations. The best ten solutions determined after 5 min had a range of density fit between the worst of the ten solutions and the best of the ten solutions of 9.5%. Three C^{α} atoms were moved about 0.5 Å while refining the side chains using grid refinement (see previous section) so as to optimize the side-chain fit; two peptide planes were fitted using grid refinement. Finally, two cycles of gradient refinement were used on the five residues, where the C^{α} atom positions were constrained for the first cycle. Although this protocol required user intervention, the total time represents about 6 min, 5 min of which was taken up using the Monte Carlo refinement.

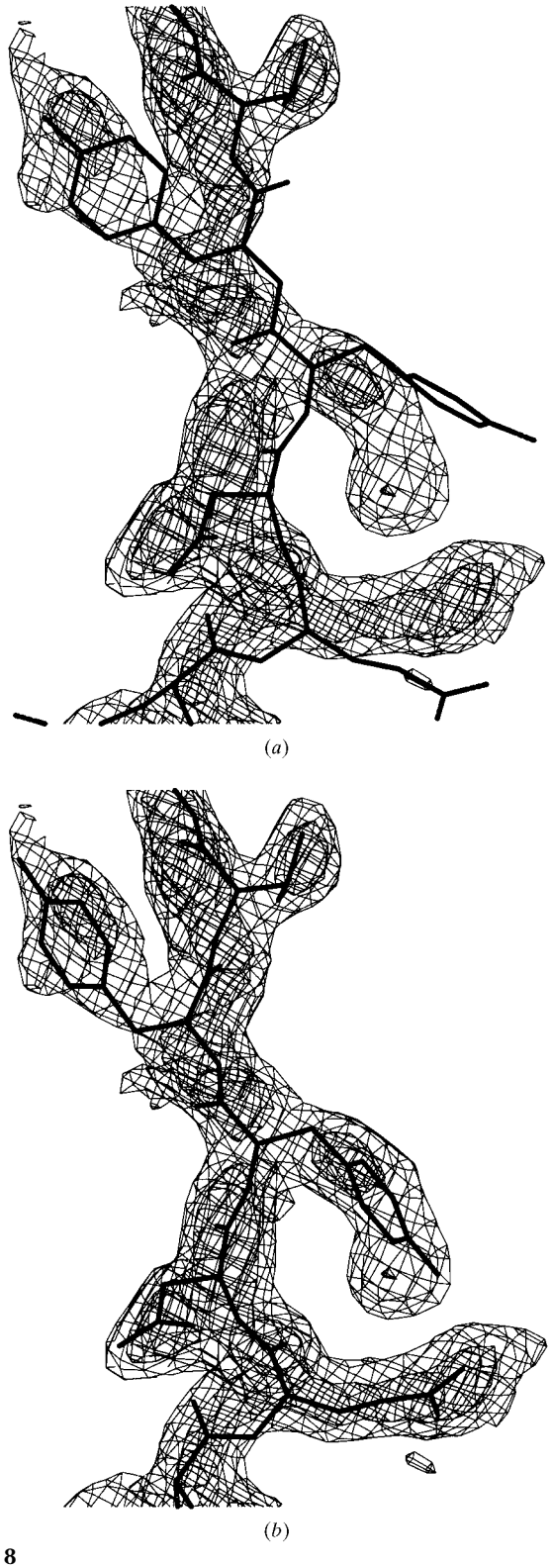

Figure 8
Real-space gradient refinement used as tool to place protein coordinates into electron density as part of model building. (*a*) The position if the protein coordinates before refinement; (*b*) the coordinates after refinement.

3.5. Refinement of water

Water structure is often ill-behaved during reciprocal-space refinement. These atoms usually have higher B factors than the protein and hence contribute very little to the higher order reflections that define atomic positions more precisely. In addition, they are not subject to bonded restraints. The manifestation of this is that water sites often move out of electron density after each reciprocal-space refinement cycle and require repositioning during the model-building phase. In addition, apparent water peaks may be the result of ripples caused by the truncation of the experimental data. This can mean that water structure appears ethereal and that first-placed solvent can often be nothing like the final water structure.

Two methods are available for the refinement of water positions in real space. The first method refines all waters

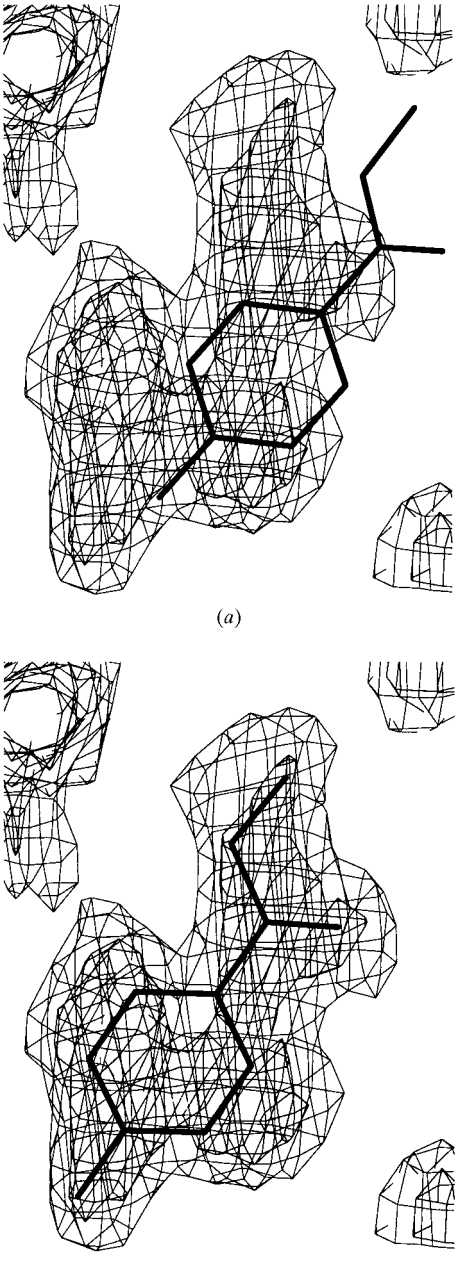

Figure 9
Gradient refinement of a ligand (methyl-*para*-benzene) into electron density. (*a*) The ligand position before refinement; (*b*) the ligand after gradient refinement of the ligand.

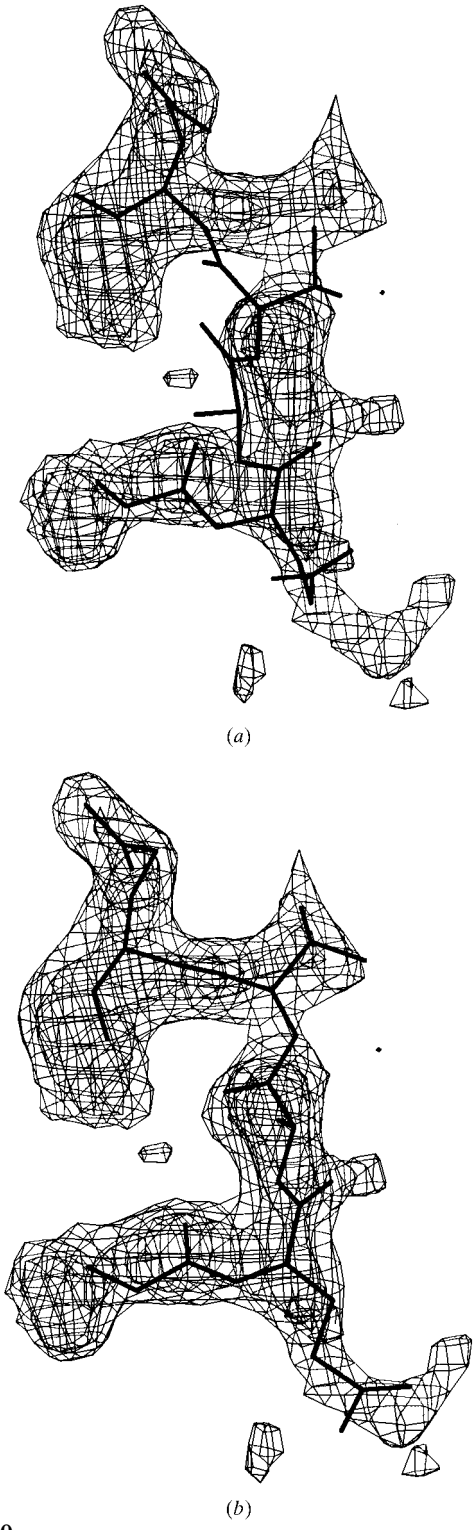

Figure 10
(*a*) Coordinates of protein atoms before a loop-fitting protocol was used to place a section of five residues out of density; (*b*) the coordinates after the protocol of Monte Carlo refinement, grid refinement and gradient refinement.

together for ten cycles with non-bond list reconstruction to all atoms between each cycle. This takes about 1 s for 200 waters. On completion of refinement each problematic water is marked using the 3D text editor (in preparation) and is flagged according to whether there is no clear gradient in the map, a bad contact, excessive shifts or poor overlap with the map. This allows large structures with many waters to be handled very rapidly and only those waters that cause problems to be shown.

The second model-building tool for water fitting places the viewer at each water position in turn, refines this site, determining non-bond residuals, map fit and map gradient and provides a summary. If none of the residuals exceed a user-defined parameter then the program moves automatically to the next water and repeats the operation. If there is a problem, the routine pauses for the user to edit the water 'manually' before continuing with the procedure.

3.6. Alternate conformations

An alternate conformation for a residue can be assumed when there is a second significant fit to density with significantly different torsion angles. The grid-refinement protocol described is not suitable for positioning alternate conformations, as it uses a tree search proceeding along a side chain until all torsion angles have been fitted. It is necessary to carry out a stack-based search over all rotatable bonds in the side chain, simultaneously storing a list of good fits to density. It is assumed that placement of alternate conformations is performed only when the structure solution is close to conclusion. In this case, there is little error in the C^{α} placement and chiral volume and no additional flexibility is needed except for the search torsion angles.

The stack-based grid search moves all side-chain torsion angles simultaneously. If there are only one or two rotatable bonds in the residue then the search has steps of 10°; if there are three torsion angles then the step size is 20° and if there are more than three the step size is 30°. It takes about 2 s to analyse either an arginine residue or a lysine residue and is essentially instantaneous for other residue types. The fit to density is determined for each conformation and the best eight fits are stored during the search. It should be noted that the search for alternate conformations can only detect three discrete ring-pucker values of atom C^{γ} for proline residues.

An alternate conformation is judged to be present if the density fit ratio for the two solutions and the sum of the differences of the side-chain torsion angles are above user-defined thresholds. The algorithm can either by used as a 'pick and add' alternate conformation tool or used as a means to search an entire molecule and place possible candidates.

4. Discussion

The aim of these refinement algorithms is to provide the crystallographer with methods that allow rapid interpretation of election density. The refinement methods have been implemented as part of tools for C^{α} tracing, generation of all atom models, general model building, H-atom placement, water refinement and ligand placement. The refinement methods described complement traditional black-box refinement programs in such a way to reduce the number of necessary cycles of refinement and model building. The methods described, along with other map-interpretation techniques (in preparation), go some way towards the full automation of map interpretation and model building for a large range of data resolutions and quality.

To date the implementation of these methods in *QUANTA* (*QUANTA*96, *QUANTA*97, *QUANTA*98) has been used to solve a large number of crystallographic structures within the YSBL. These range from *de novo* structure determinations (MAD, SIR and MIR) as well as MR structure determinations from a resolution of 0.95 to 3.0 Å. The program *QUANTA* is available under license from MSI.

I would like to thank the members of the York Structure Biology group for both testing the program and providing ideas that were implemented. I would like to thank Eleanor Dodson and Garib Murshudov for help with writing this paper, and Eleanor Dodson, Garib Murshudov and Kevin Cowtan for discussions on mathematics and crystallography. This work was entirely funded by Molecular Simulations Inc.

References

Adams, P. D., Pannu, N. S. & Read, R. J. (1997). *Proc. Natl Acad. Sci. USA*, **94**(10), 5018–5023.
Booth, A. D. (1946). *Proc. R. Soc. London Ser. A*, **188**, 77–92.
Brent, R. P. (1973). *Algorithms for Minimization without Derivatives*, ch. 5. Englewood Cliffs, NJ: Prentice-Hall.
Bricogne, G. (1993). *Acta Cryst.* D**49**, 37–60.
Bricogne, G. (1988). *Acta Cryst.* A**44**, 517–545.
Brünger, A. T. (1992). *X-PLOR Manual, Version 3.1. A System for X-ray Crystallography and NMR.* Yale University, Connecticut, USA.
Carter, C. W. Jr, Crumley, K. V., Coleman, D. E., Hage, F. & Bricogne, G. (1990). *Acta Cryst.* A**46,** 57–68.
Chen, Z., Blanc, E. & Chapman, M. S. (1999). *Acta Cryst.* D**55**, 464–468.
Cochran, W. (1948). *Acta Cryst.* **1**, 138–142.
Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* D**50**, 760–763.
Cowtan, K. (1994). *Jnt CCP4/ESF–EACBM Newslett. Protein Crystallogr.* **31**, 34–38.
Diamond, R. (1971). *Acta Cryst.* A**27**, 436–452.
Engh, R. A. & Huber, R. (1991). *Acta Cryst.* A**47**(4), 392–400.
Fletcher, R. (1987). *Practical Methods of Optimization.* 2nd ed. Chichester: Wiley.
Goffeau, A., Barrell, B. G., Bussey, H., Davis, R. W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J. D., Jacq, C., Johnston, M., Louis, R. J., Mewes, H. W., Murakami, Y., Philippsen, P., Tettelin, H. & Oliver, S. G. (1996). *Science*, **274**, 546–567.
Hughes, E. W. (1941). *J. Am. Chem. Soc.* **63**, 1737–1752.
Jones, J. A. (1985). *Methods Enzymol.* **115**, 157.
Jones, T. A. & Kjeldgaard, M. (1997). *Methods Enzymol.* **227**, 173–230.
Jones, T. A., Zou, J. Y., Cowan, S. W. & Kjeldgaard, M. (1991). *Acta Cryst.* A**47**, 110.
Knuth, D. E. (1981). *Seminumerical Algorithms*, Vol. 2, 2nd ed., Reading, MA, USA: Addison-Wesley.

Konnert, J. H. & Hendrickson, W. A. (1980). *Acta Cryst.* A**36**, 344–350.
Lunin, V. Yu. & Urzhumtsev, A. G. (1984). *Acta Cryst.* A**40**, 269–277.
McRee, D. E. (1999). *J. Struct. Biol.* **125**(2–3), 156–165.
Murshudov, G. N., Vagin, A. A. & Dodson, E. J. (1997). *Acta Cryst.* D**53**(3), 240–255.
Navaza, J. & Saludjian, P. (1997). *Methods Enzymol.* **276**, 581–594.
Oldfield, T. J. (1996). *Proceedings of the CCP4 Study Weekend. Macromolecular Refinement*, edited by E. Dodson, M. Moore, A. Ralph & S. Bailey, pp. 67–74. Warrington: Daresbury Laboratory.
Pannu, N. S. & Read, R. J. (1996). *Acta Cryst.* A**52**, 659–668.
Perrakis, A., Morris, R. & Lamzin, V. S. (1999). *Nature Struct. Biol.* **6**(5), 459–463.
Ramachandran, G. N. & Saisiskharan, V. (1969). *Adv. Protein Chem.* **23**, 283–437.
Read, R. J. (1986). *Acta Cryst.* A**42**, 140–149.
Read, R. J. (1990). *Acta Cryst.* A**46**, 900–912.
Rice, L. M. & Brünger, A. T. (1994). *Proteins Struct. Funct. Genet.* **19(4)**, 277–290.
Rice, S. O. (1954). *Selected Papers on Noise and Stochastic Processes*, edited by N. Wax, pp. 133–195. New York: Dover.
Sack, S. & Quiocho, F. A. (1997). *Methods Enzymol.* **227**, 158–173.
Sevcik, J., Dodson, E. J. & Dodson, G. G. (1991). *Acta Cryst.* B**47**, 240–353.
Sheldrick, G. M. & Schneider, T. R. (1997). *Methods Enzymol.* **227**, 319–343.
Stuart, A. & Ord, K. J. (1991). *Kendall's Advanced Theory of Statistics*, Vol. 2, 5th ed. London/Melbourne/Auckland: Edward Arnold.
Surles, M. (1994). *Protein Sci.* **3**, 198–210.
Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**(4), 849–861.
Torn, A. & Zilinskas, A. (1987). *Global Optimization: Lecture Notes in Computer Science.* Berlin: Springer-Verlag.
Tronrud, D. E., Ten Eyck, L. & Matthews, B. W. (1987). *Acta Cryst.* A**34**, 489–501.
Turk, D. (1992). PhD thesis. Technische Universität München, Germany.
Vagin, A. A. & Telplyakov, A. (1997). *J. Appl. Cryst.* **30**(6), 1022–1025.
Weeks, C. M., De Titta, G. T., Hauptman, H. A., Thuman, P. & Miller, R. (1994). *Acta Cryst.* A**50**, 210–220.
Whittingham, J. L., Chaudhuri, S., Dodson, E. J., Moody, P. C. E. & Dodson, G. G. (1995). *Biochemistry*, **34**, 15553–15563.