Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2009
August 30 - September 2, 2009, San Diego, California, USA

DETC2009-87737

TORSION ANGLE REFINEMENT AND DYNAMICS AS A TOOL TO AID CRYSTALLOGRAPHIC STRUCTURE DETERMINATION

Ralf W. Grosse-Kunstleve
Nigel W. Moriarty
Paul D. Adams
Lawrence Berkeley National Laboratory
Berkeley, CA 94720, USA

ABSTRACT

Crystallographic methods using experimental diffraction data have produced about 85% of the macromolecular structures in the Protein Data Bank. Before deposition, nearly all crystal structures are refined with gradient-driven optimization techniques. Refinement is typically performed with iterative local optimization methods. A common problem is convergence to local minima. Reparameterization of the model in torsion angle space reduces the number of parameters. This in itself can help to escape from local minima. Combination with rigid-body dynamics algorithms results in an important tool for sampling conformational space. This paper presents the torsion angle refinement and dynamics algorithms implemented for the phenix.refine program and the results of various tests.

INTRODUCTION

The rapid growth of known macromolecular structures, commonly deposited in the Protein Data Bank (PDB) (1), has been fueled in large part by the use crystallographic methods. Crystals of macromolecules are subject to X-ray or neutron radiation to produce diffraction data. These experimental data are the input for diverse, complex structure determination methods that yield an approximate, initial structural model. Eventually, all models are subject to *refinement*, which is the process of optimizing the fit of diffraction data calculated directly from the model to the experimental observations.

Early crystallographic refinement procedures used least-squares methods. Today, more sophisticated maximum-likelihood methods are in wide use (2). Both approaches can be combined with iterative gradient-driven minimization methods, using gradients of the least-squares or maximum-likelihood function with respect to the model parameters. Improved estimates of the model parameters are commonly obtained with quasi-Newton or conjugate-gradients methods (3). Cycles of model updates and gradient calculations are repeated until the fit is of sufficient quality to answer the scientific questions motivating the experiment.

In conventional refinement, the number of model parameters is a direct linear function of the number of atoms in the macromolecule. For example, when refining Cartesian atomic coordinates, the number of model parameters is three times the number of atoms. The number of atoms in macromolecules ranges from a few hundred to hundreds of thousands (the median number of atoms in the PDB is about 2900 atoms). Parameter spaces of this size preclude the use of global optimization techniques; therefore iterative local optimization methods, as outlined above, are typically used. This implies that the gradient-driven methods converge from the starting estimate of the model parameters to the next local minimum. Almost universally, prior chemical knowledge has to be included in the target function in order to arrive at models that both fit the data and make chemical sense. To this day, generally time-consuming manual intervention is essential.

Another important reason for using prior chemical knowledge is that the resolution of the experimental data is often limited. For macromolecular structures, atomic-resolution data (around 1.0 Å) are available only in exceptional cases. Resolutions around 2.5 Å are more usual, and resolutions of 3.5 Å are still common. Some biologically important structures were refined with significantly lower resolution data, for example the ribosome (4).

Torsion angle dynamics was first introduced into the context of macromolecular structure refinement by Rice and Brunger (5). A major motivation was to aid refinement of low-resolution structures. In torsion angle space, the parameter-to-observation ratio is reduced by a factor of approximately seven compared to a model with Cartesian atomic coordinates, or even more if certain torsion angles are fixed based on chemical

knowledge. Rice and Brunger used torsion angle dynamics combined with simulated annealing as a tool to escape from local minima encountered in gradient-driven minimization.

Our interest in torsion angle refinement and dynamics is connected to the development of phenix.refine, the refinement module of the PHENIX project (5). Integration of torsion angle parameterization in the existing suite of phenix.refine algorithms is expected to enhance the capabilities for low-resolution refinement. We also anticipate benefits simply from moving between Cartesian and torsion angle refinement, since the energy landscapes as seen by the minimizer have different local minima.

In this paper, we describe our implementation of the dynamics engine developed for phenix.refine. It is based on the recursive Articulated-Body Forward Dynamics algorithm of Featherstone (7). A simplified version of Featherstone's recursive Inverse Dynamics algorithm (7) is used for torsion angle gradient-driven refinement.

The second major component we describe is the tree-generation step required for the recursive algorithms. This amounts to a *rigidity analysis*, assuming that all bonds and bond-angles are fixed. A comprehensive mathematical treatment of this problem and an outline of a fast, patented "pebble-game" algorithm were published by Jacobs (8, 9). We chose to use a different approach to the tree generation, which is also fast and works correctly for the vast majority of practical cases.

In the following we adopt the nomenclature and notation of Featherstone (7).

RECURSIVE DYNAMICS ALGORITHMS

A comprehensive overview of the history of recursive dynamics algorithms was given by Mukherjee and Anderson (11). Recently, Featherstone published a very detailed treatment of this subject (7), along with MATLAB sources of key algorithms (12). Our work is based directly on these sources and uses the nomenclature of Featherstone throughout. Our main addition to Featherstone's work is an object-oriented framework for a joint library, and the algorithms for connecting an atomic macromolecular model to the core dynamics algorithms. This includes algorithms for converting gradients with respect to Cartesian atomic coordinates, as they were available already in our existing phenix.refine framework, to *external forces* suitable for the rigid body dynamics algorithms. The underlying equations can be found in Schwieters and Clore (13), which in turn is heavily based on work by Jain, Vaidehi and Rodriguez (14). Torsion angle minimization is implemented using the recursive Inverse Dynamics algorithm of Featherstone (7) with all velocities set to zero. We are using a simplified version of the algorithm in which all calculations involving velocities and inertia matrices are removed.

For uniformity, in our application we work exclusively with external forces, but the ability of the original algorithms to also handle internal forces is preserved.

Our reference implementation of the complete dynamics engine makes use of the high-level Python programming language (15). The self-contained result is very compact and organized in a reusable way (16).

RIGIDITY ANALYSIS AND TREE GENERATION

Recursive dynamics algorithms require a spanning tree structure of all the rigid bodies in the system. This is described in great detail by Featherstone (7). To generate such a spanning tree for a molecular structure, we abstract the bonding model to a mathematical graph of vertices (representing the atoms) and edges (representing the bonds).

The torsion angle parameterization implies that all bonds and bond-angles are fixed, leaving torsion angles as the only remaining degrees of freedom. We approached the rigidity analysis and tree generation from an equivalent view of *rotatable bonds*. In an acyclic tree, all bonds are rotatable. However, if there are loops, as found in most molecular structures, certain bonds are no longer rotatable. Groups of vertices form rigid clusters in which all distances between all vertices in the cluster are fixed. The simplest examples of (sub)sets of vertices forming rigid clusters are loops of up to six vertices. This follows, for example, directly from the work of Jacobs (8). Larger loops of size N have N-6 de-localized degrees of freedom.

Systematic evaluations of *rigidity matrices* (17) lead us to a simple but powerful observation. It applies to the situation of two *pivot* vertices connected by three arches, as illustrated in Fig. 1. Let N_i be the number of vertices in each arch i (excluding the pivot vertices), with i={0,1,2} and $N_i > 0$. We found that a configuration of three arches is a rigid cluster only if $N_i < 6$ for each i, and the sum of $N_i < 10$.

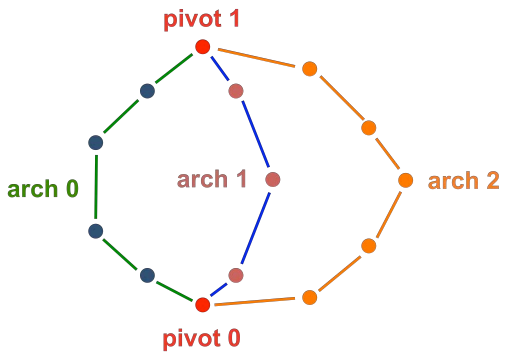

Figure 1. Illustration for the 3-arch rigidity rules. This example configuration is not rigid because the sum of the vertices in the three arches (4+3+5) is not less than 10.

Using the observation, we define rigid clusters by searching for rigid loops and rigid three-arch configurations. Abstracting each rigid cluster to a meta-vertex, and reducing the original edges accordingly, we repeat the procedure, until no new clusters are found. We applied this algorithm to a library of about 8 million drug-like compounds obtained from the ZINC database (18). For all these compounds, the rigid clusters are determined correctly, as verified by comparison with rigidity matrix calculations. We also tested with a large

subset of the CCDC database (19), limited to structures with less than 80 atoms, since the rigidity matrix verification is very slow for larger structures. We found only 71 out of 106835 cases with missed rigid clusters.

When building the spanning tree of rigid clusters, large-scale loops, such as loops due to disulfide bonds in proteins, present special challenges (11). This is true even if there are no missed rigid clusters. Our approach is to cut large-scale loops, and to replace the constraints with harmonic potentials restraining the bond distances and bond angles. In our context, these potentials are readily available since they are widely used in *restrained refinement* (2).

In the rare cases of missed rigid clusters, there will be a few additional restraints that could be handled as constraints.

TORSION ANGLE REFINEMENT AND DYNAMICS MODEL

In our implementation, the Featherstone system model (7), the result of the rigidity analysis and tree generation, and a *potential object*, are managed by a *TARDY model*, short for Torsion Angle Refinement and Dynamics model. The TARDY model provides methods for computing the moved Cartesian coordinates x_{moved} of the mass points, the kinetic energy E_{kin}, potential energy E_{pot}, and the generalized accelerations $\ddot{q}$. Calculation of the potential energy and gradients of the potential energy w.r.t. Cartesian coordinates is delegated to the potential object. In this way, the TARDY model can be reused with different potentials. Currently implemented are:

- A simple *reference potential object* for quick testing, which restrains the Cartesian coordinates x_{moved} to reference coordinates $x_{reference}$. The harmonic restraints are of the form $(x_{moved} - x_{reference})^2$; the potential energy is the sum of these terms over all mass points.
- A *real-space potential object* combining a *real-space target function* and geometry restraints (2, 20).
- A *reciprocal-space potential object* combining various standard reciprocal-space target functions (21) and geometry restraints.

The TARDY model also provides methods for the assignment of initial velocities, velocity scaling to obtain a given kinetic energy, performing a dynamics time step, and performing gradient-driven minimization using the L-BFGS quasi-Newton minimizer (3). The TARDY model automatically keeps track of all interdependencies of the calculations involved, maximizing ease of reusability and protection against inadvertently introducing inconsistencies.

Our procedure for performing the integration for a dynamics step is the simplest possible, essentially (in this order):

time step positions: $q_{t+\Delta t} = q_t + \dot{q}\Delta t$ (1)

time step velocities: $\dot{q}_{t+\Delta t} = \dot{q}_t + \ddot{q}\Delta t$ (2)

The velocity time step uses Eq. 2 exactly for all joint types. The positional time step employs Featherstone's Equation 4.3 for the spherical joint and the rotational components of the 6-DoF joint. For our purposes, we did not find it necessary to explore more sophisticated integration algorithms.

The assignment of initial velocities is based on the simple idea to scale the generalized velocities $\dot{q}$ of each joint according to the mass that is moved by the joint. For this, the accumulated spatial inertia matrix for each joint is computed by traversing the spanning tree from the leaves to the root via

$$I_i^{accu} = I_i + \sum_{j \in \mu(i)} {}^{i}X_j^{*} I_j^{accu\ j}X_i \qquad (3)$$

similar to the first equation in Pass 2 of Table 7.1 of (7). In a loop over all joints, each $\dot{q}_{i,k}$ element k of joint i is set to unity in turn, with all other elements held at zero. The corresponding $\dot{q}_{i,k}^{scale}$ factor is obtained via:

$$\dot{q}_{i,k}^{scale} = 1/\sqrt{E_{kin}} \qquad (4)$$

with:

$$\begin{aligned} E_{kin} &= v_i I_i^{accu} v_i \\ v_i &= S_i \dot{q}_i \end{aligned} \qquad (5)$$

In the next step, generalized velocities are assigned by randomly drawing from Gaussian distributions with standard deviations $\dot{q}_{i,k}^{scale}$. The kinetic energy E_{kin}^{random} of the entire system obtained in this way fluctuates around unity. An exact desired kinetic energy E_{kin}^{target} can be obtained by multiplying all generalized velocities with the factor:

$$\sqrt{E_{kin}^{target} / E_{kin}^{random}} \qquad (6)$$

We note that the procedure for obtaining $\dot{q}^{scale}{}_{i,k}$ is equivalent to setting all $\dot{q}$ in the entire system to zero, except for $\dot{q}_{i,k}$ which is set to unity, performing a positional time step with $\Delta t = 1$, computing the kinetic energy of the entire system, which is then substituted into Eq. 4.

For gradient-driven minimization, gradients w.r.t. the generalized positional coordinates q are required. For the translational components of the 6-DoF joint and for the revolute joint the expressions for the gradients are straightforward. For the spherical joint and the rotational components of the 6-DoF joint the expressions are more involved. If Euler angles are used, expression found in (7) and (22) can be reused. We are not aware of literature with gradient expressions for Euler parameters. An essential step is to handle non-normalized Euler parameters as produced by the minimizer. For this, gradients w.r.t. non-normalized Euler parameters need to be converted to gradients w.r.t. normalized Euler parameters. By using the chain

rule, we arrived at this coefficient matrix C_{pg} for the gradient conversion:

$$C_{pg} = \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,1} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{0,2} & c_{1,2} & c_{2,2} & c_{2,3} \\ c_{0,3} & c_{1,3} & c_{2,3} & c_{3,3} \end{bmatrix} / a^3 \qquad (7)$$

With:

$$\begin{aligned} c_{0,0} &= p_1^2 + p_2^2 + p_3^2 \\ c_{1,1} &= p_0^2 + p_2^2 + p_3^2 \\ c_{2,2} &= p_0^2 + p_1^2 + p_3^2 \\ c_{3,3} &= p_0^2 + p_1^2 + p_2^2 \\ c_{0,1} &= -p_0 p_1 \\ c_{0,2} &= -p_0 p_2 \\ c_{0,3} &= -p_0 p_3 \\ c_{1,2} &= -p_1 p_2 \\ c_{1,3} &= -p_1 p_3 \\ c_{2,3} &= -p_2 p_3 \\ a &= \sqrt{p_0^2 + p_1^2 + p_2^2 + p_3^2} \end{aligned} \qquad (8)$$

Here the p_i are the non-normalized parameters.

TESTS USING THE REFERENCE POTENTIAL OBJECT

The main purpose of the reference potential object is to facilitate tests ensuring that all parts of the TARDY model implementation are working properly. The primary measures of quality exercised are:

- Agreement of analytical gradients and finite difference gradients of the potential energy w.r.t. generalized coordinates. This straightforward test is integrated in trial gradient-driven minimizations of a variety of small molecular structures.
- Conservation of total energy over 10^5 dynamics steps, using different time step sizes.

The reference potential object also proved useful for comparing Euler parameter vs. Euler angle parameterizations of the rotational components of the 6-DoF joint, and for comparing Featherstone's parameterization of the translational components of the 6-DoF joint with a simpler alternative. Featherstone sets

$$r = E^{-1}\left[q_5, q_6, q_7\right]^T \qquad (9)$$

(see Table 4.1 in (7)). The simpler alternative we explored is:

$$r = \left[q_5, q_6, q_7\right]^T \qquad (10)$$

Fig. 2 shows the results of our comparisons of dynamics simulations of the motion of a 6-DoF body. The green and the blue plot employ Eq. 9, while the black and the red plots employ Eq. 10. Evidently, the energy is better conserved if Eq. 10 is used. The accuracy problem of Eq. 9 is actually mentioned in Example 4.5 of (7). Therefore we decided to use Eq. 10 in all our following work. We note that this is the only point at which our implementation deviates from the algorithms as presented in (7).

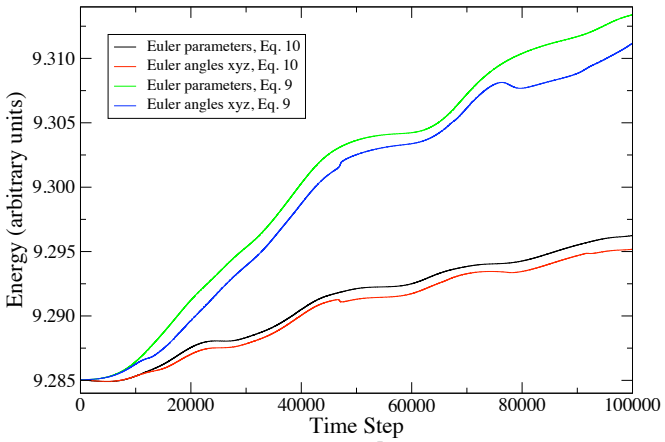

Figure 2. Energy conservation over 10^5 dynamics steps, Euler angles vs. Euler parameters, and Eq. 9 vs. Eq. 10.

The black and the green plot in Fig. 2 employ Euler parameters, while the red and the blue plot employ Euler angles; the equations used can be found in section 4.5 of (7). The motivation to implement both was mutual validation: if two different approaches deliver meaningful and very similar results, it is safe to conclude that both are correct. Clearly, Fig. 2 validates our implementations. It was a small surprise to see that the energy conservation is slightly better for Euler angles. This observation persisted when reviewing results obtained with other random seeds. However, the difference is much less significant than that between Eqs. 9 and 10. Furthermore, the plots employing Euler angles show discontinuities that we attribute to the well-known gimbal lock problem (see e.g. Wikipedia).

Fig. 3 shows the results of a test series for gradient-driven minimization of the orientation of a 6-DoF body; in the crystallographic literature this is known as *rigid-body refinement* (21, 23). In addition to the *xyz* Euler angle convention of (7), we also explored the *zxz* convention of (22) and two unusual 4-Euler-angle conventions, *xyzy* and *yxyz*, with redundant parameters, like Euler parameters, but composing rotation matrices, like the usual Euler angle conventions. In the notation of (7):

$$\begin{aligned} E_{xyzy} &= rx(q_4)ry(q_3)rz(q_2)ry(q_1) \\ E_{yxyz} &= ry(q_4)rx(q_3)ry(q_2)rz(q_1) \end{aligned} \qquad (11)$$

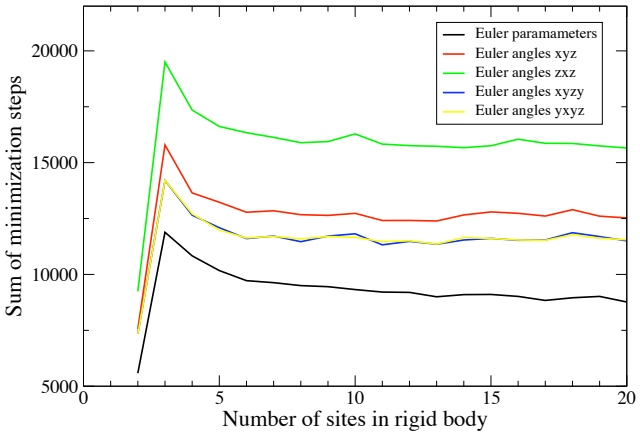

Figure 3. Euler parameters vs. angles, gradient-driven minimization. The vertical axis shows the sum of the minimization steps needed to achieve convergence in 1000 trials with random displacements. The horizontal axis shows the number of sites in the rigid body. The plots show that the size of the rigid body does not have a profound influence, and that the results can therefore be generalized.

For each parameterization, these steps were carried out:

- Loop over number of sites: 2-20
  (horizontal axis in Fig. 3)
- 1000 trials each:
- random sites → original orientation
- random rotation → displaced orientation
- harmonically restrain displaced to original orientation
- refine back to original orientation using L-BFGS minimizer (3)
- report total number of minimization steps (vertical axis in Fig. 3)

A smaller total number of minimization steps means that the refinement procedure is more efficient. The results in Fig. 3 mirror results reported previously (23) in that the *xyz* Euler angle convention is better suited for rigid body refinement than the *zxz* convention (or any other convention with the first and third rotation around the same axis). This suggests that our simple test system is representative of more complex real-world applications and thereby supports the significance of our main observation: Euler parameters are the best choice for rigid-body refinement. This is a new result, as we are not aware of other rigid-body refinement programs using Euler parameters.

Interestingly, our experimental *xyzy* and *yxyz* 4-Euler-angle conventions also perform better than the usual *xyz* convention, but not as well as Euler parameters.

Considering all results presented in this section, we decided to use Euler parameters and Eq. 10 in all our following work, for both dynamics and gradient-driven minimization.

In a final validation of the energy conservation we combined a 6-DoF body with a second body connected via a revolute joint. The kinetic energy at the start was set to unity with random velocities (Eqs. 3-6). The potential energy at the start was zero. Fig. 4 shows example plots of the kinetic, potential, and total energy for time step size 10^{-4}. Tab. 1 shows the values of the total energy after 10^5 time steps as a function of the time step size. As expected, the dynamics simulation is unstable if the time step is too large. With suitable time step sizes, the total energy is conserved very well.

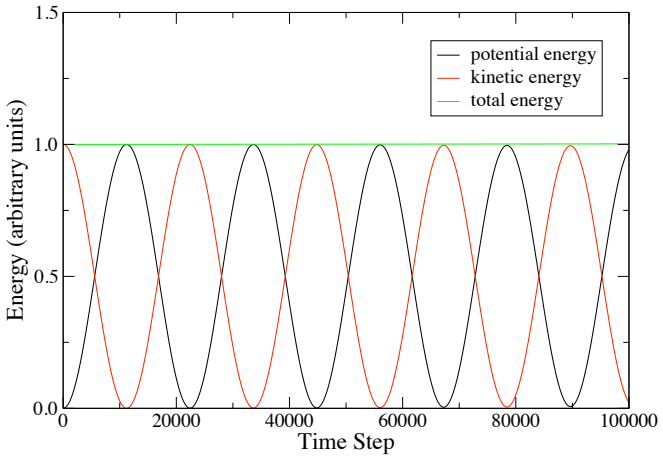

Figure 4. Example plots of kinetic, potential, and total energy over 10^5 time steps, with time step size 10^{-4}. 6-DoF body with joint using Euler parameters and Eq. 10.

time step size (arbitrary units)	total energy after 10^5 steps
10^{-6}	1
10^{-5}	1.00002
10^{-4}	1.00199
10^{-3}	1.22344
10^{-2}	numerical error (2516.81 after 31000 steps)

Table 1. Total energy after 10^5 dynamics steps as a function of the time step size. 6-DoF body connected to second body via a revolute joint.

TESTS USING THE REAL-SPACE POTENTIAL OBJECT

Central to the real-space potential object is a 3-dimensional array of values proportional to the electron density in a crystallographic unit cell, often called an *electron density map*. In the actual refinement against experimental diffraction data only an approximation of the electron density is known (21). Typically, likelihood-weighted maps are used instead of actual electron density maps. For the purpose of systematic testing presented here, we decided to use idealized electron density maps, but computed with the same process as used for the approximate maps: first, diffraction data are predicted using the atomic model, then a Fourier transformation is carried out to obtain the real-space map from the diffraction data. This process is described in standard references, e.g. (21, 24). The real-space map is typically sampled at ⅓ of the resolution of the diffraction data. Based on our previous experience, we decided to work with this widely used factor without exploring alternatives.

The potential function E_{rs} associated with the real-space map is simply the sum of the density values at the current positions of the atomic centers. The gradients of the potential function are obtained with the finite-difference method: each

atom is moved a certain distance $\pm d_{fd}$ parallel to the basis vectors of the Cartesian system. The gradients are the differences of the density values at the two points $\pm d_{fd}$ away from the atomic center, divided by $2d_{fd}$. The density values are obtained via standard linear eight-point interpolation, using the eight nearest neighbors in the real-space map.

The potential function used in the tests is the weighted sum of the real-space potential E_{rs} and a potential E_{geo} due to geometry restraints:

$$E_{pot} = w_{rs}E_{rs} + E_{geo} \qquad (12)$$

The geometry restraints are in turn the sum of harmonic bond, angle, chirality, and planarity restraints, harmonic or sinusoidal dihedral restraints, and repulsive nonbonded interactions (2, 20, 25).

Our tests with the real-space potential cover a systematic exploration of the behavior of gradient-driven minimization and simulated annealing (5). The parameter space explored in both cases includes:

- The resolution of the synthetic diffraction data. In actual applications, it is dictated by the experiment. We sampled four resolutions: 1.25 Å, 2.5 Å, 3.75 Å, 5 Å.
- The real-space target weight w_{rs} of Eq. 12.
- The value of d_{fd} for the computation of real-space gradients.
- Random displacements of the atomic model, both in Cartesian space and in torsion-angle space.

In a set of preliminary tests, we observed that the target weight w_{rs} is the single most important parameter. We tried the values w_{rs} = 1,10,30,50,100,200,250,500,1000,2000. After reviewing the preliminary results, we ran the final set of tests with the selected values w_{rs} = 10,100,500, as these sufficiently represent all outcomes.

We observed that the value of d_{fd} is not very critical. We sampled values $\frac{1}{3},\frac{1}{2},\frac{2}{3},1$ of the resolution of the diffraction data. For the final tests we settled on the factor $\frac{1}{3}$ (which is the same factor used for the sampling of the real-space map).

We implemented two methods for the generation of random displacements of the atomic models, relative to the atomic models used for computing the idealized real-space maps: a method in Cartesian space, and a method in torsion-angle space. In the Cartesian space method, atomic coordinates are first moved randomly parallel to the basis vectors of the Cartesian system. With this approach it is straightforward to obtain a given target root-mean-square displacement (RMSD) of the coordinates, but the potential energy due to the resulting geometry restraints generally increases to implausible values. This is corrected in a second stage by up to 500 steps of minimization of the geometry potential alone. The torsion-angle space method for generating random displacements is a short dynamics simulation starting with random velocities and a small time step. The time step is gradually increased until the RMSD of the coordinates is above a certain lower threshold. If the RMSD increases beyond a certain upper threshold in the last step, the step is discarded and repeated with a smaller time step, until the RMSD falls into the range between the lower and upper threshold. The target RMSD values used in our tests are $\frac{1}{3},\frac{2}{3},1$ times the resolution of the synthetic diffraction data, representing modest to very large deviations. The actual RMSD values obtained with both methods outlined above are distributed around the target values.

In the simulated annealing (SA) tests, these additional parameters were explored:

- The start temperature. In preliminary tests we tried 2500 K, 5000 K, 10000 K. After reviewing the results we chose 2500 K and 5000 K for the final set of tests.
- The number of cooling steps to reach the final temperature of 300 K. The values used are 250 and 500.

The temperature is decreased linearly after each step. Velocity scaling (Eq. 6) is used to enforce the target temperature. The conversion between temperature and kinetic energy is given by

$$temperature = \frac{E_{kin}}{\frac{1}{2}kd} \qquad (13)$$

where $k = 0.0488878$ is the Boltzmann constant in AKMA units (25), a system of units compatible with the parameterization of the geometry restraints. d is the number of degrees of freedom, which we set to three times the number of atoms in both Cartesian space and torsion-angle space, in order to achieve a universal mapping between kinetic energy and temperature.

All tests outlined above were run in both Cartesian space and torsion-angle space, but using the same general implementation: for the torsion angle space runs, the rigidity-analysis and tree generation procedure was used to define rigid bodies connected by revolute joints to a 6-DoF base body; for the Cartesian space runs, each atom was a separate body with a 3-DoF translational joint.

Four small poly-peptide models were prepared for the tests. The smallest model (GLY-GLY box) consists of two glycines embedded in an artificial unit cell constructed around the molecule, with a 5 Å surrounding buffer. Similarly, the next larger model (LYS-PRO-TPR box) is a chain of lysine-proline-tryptophan in an artificial unit cell. The third model (1YJP box) is PDB entry 1YJP but in an artificial unit cell and with the water molecules deleted. The fourth model (1YJP cryst.) is the same molecule in the crystallographic unit cell with space group $P2_1$ (26). The three "box" models are designed to exercise the ability of the minimization and SA procedures to pull the randomly displaced models back into the original configuration, guided by the real-space map, without

interference from neighboring molecules (hence the 5 Å buffer). The 1YJP cryst. model was chosen to explore the influence of neighboring density.

The total number of final minimization tests run was 6912:
4 models
x 4 resolutions
x 3 weights
x 2 parameter spaces (Cartesian, torsion angle)
x 3 target RMSD
x 2 methods of random displacements
x 12 random seeds

The total number of final SA tests run was 27648:
6912 as before
x 2 start temperatures
x 2 values for the number of cooling steps

The time step used in all SA tests was 1 femto second. Due to the continuous velocity scaling, this value is not very critical.

The results of the tests were evaluated by inspection of start RMSD vs. final RMSD scatter plots. Each plot includes the results of 36 tests: 3 target RMSD x 12 random seeds. All 960 plots are archived for review (see Appendix). Fig. 5 shows selected plots as a typical example for the influence of the real-space weight w_{rs} in SA runs. The weight w_{rs} = 10 leads to many final RMSD significantly larger than the start RMSD. The weights w_{rs} = 100 and 500 both give significantly better results. Comparison of the torsion-angle and Cartesian SA plots in Fig. 5 shows that in this case torsion-angle SA clearly outperforms equivalent runs in Cartesian space. Fig. 6 shows a counter example as observed mostly with low-resolution data. In this situation the real-space map provides only limited guidance. Fig. 6 illustrates the tendency of Cartesian SA to essentially not move the model, and the tendency of torsion-angle SA to move the model further away from the target configuration. Tab. 2 shows a more global representation of the test results, excluding tests with $w_{rs} = 10$ for reasons apparent from Fig. 5. The primary motivation for the design of Tab. 2 is to deliniate the situations in which torsion-angle SA outperforms Cartesian SA. For this, two measures were used: (a) the smallest final RMSD found in each plot and (b) the mean final RMSD. The values for corresponding torsion-angle and Cartesian plots were compared. The "tt" row in Tab. 2 shows the number of times both measures are smaller in the torsion-angle plot, "tc" shows the number of times the smallest final RMSD value is smaller in the torsion-angle plot, but the mean is smaller in the Cartesian plot, "ct" is vice versa, and "cc" shows the number of times both measures are better in the Cartesian plot. The counts are separated by model and type of random displacement procedure, but are the sum for all other parameters. I.e. the table can be understood to reflect overall expectations averaged over resolutions, real-space weights, start temperatures, and number of cooling steps. The main pattern that emerges from Tab. 2 is that torsion-angle SA outperforms Cartesian SA for the smaller models, but that it can be the other way around for larger models, in particular if there is neighboring density (1YJP cryst.). Tab. 3 was prepared in a manner very similar to Tab. 2, but comparing results of the minimization test runs. The main pattern is very similar in both tables.

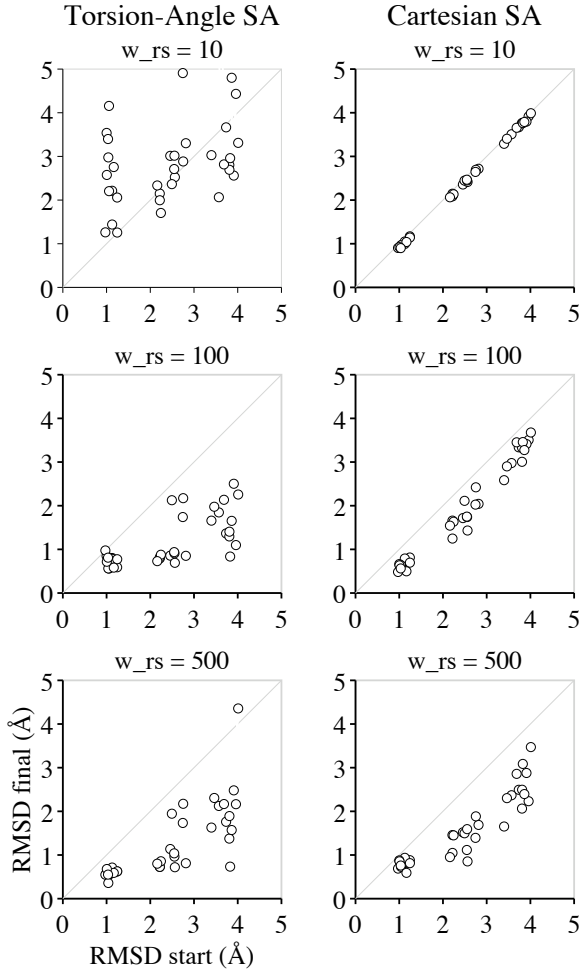

Figure 5. Results of Simulated Annealing tests. Model: 1YJP (box), "w_rs" is the real-space weight w_{rs}, resolution of diffraction data: 3.75 Å, random displacements in torsion-angle space, start temperature 5000 K, 500 cooling steps.

CONCLUSION

The first set of test results presented here, based on the simple reference potential object, establishes that our implementation of the core torsion angle refinement and dynamics algorithms uses the best available parameterizations (Euler parameters, Eq. 10), produces correct analytical gradients for minimization, and correctly conserves the total energy in dynamics (Fig. 4). The second set of tests, based on the real-space potential object hint that it could be beneficial to subdivide large molecules into smaller rigid units, e.g. groups of three peptides, for torsion-angle minimization or SA, and to use standard geometry restraints to maintain connectivity between the units. We are planning to explore this idea, in

addition to integrating our procedures for more traditional torsion-angle SA in the phenix.refine program.

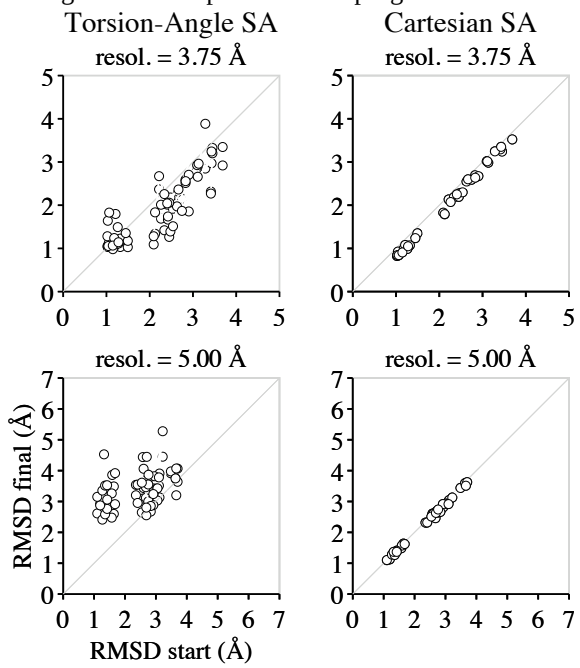

Figure 6: Results of Simulated Annealing tests. Model: 1YJP (cryst.), "resol." is the resolution of diffraction data, real-space weight: w_{rs}=100, random displacements in Cartesian space, start temperature 5000 K, 500 cooling steps.

	GLY-GLY (box)		LYS-PRO-TRP (box)		1YJP (box)		1YJP (cryst.)	
rand. displ.	t.a.	cart.	t.a.	cart.	t.a.	cart.	t.a.	cart.
tt	26	22	28	17	16	5	15	5
tc	0	4	0	1	6	0	0	0
ct	6	6	4	6	9	14	7	13
cc	0	0	0	8	1	13	10	14

Table 2. Global summary of Simulated Annealing results (weight 10 omitted). See text for details.

	GLY-GLY (box)		LYS-PRO-TRP (box)		1YJP (box)		1YJP (cryst.)	
rand. displ.	t.a.	cart.	t.a.	cart.	t.a.	cart.	t.a.	cart.
tt	5	3	5	1	3	1	1	0
tc	3	4	1	3	1	1	3	2
ct	0	0	0	0	0	0	0	0
cc	0	1	2	4	4	6	4	6

Table 3. Global summary of minimization results (weight 10 omitted). See text for details.

ACKNOWLEDGMENTS

We acknowledge financial support from NIH-NIGMS under grant No. P01GM063210 and support from the US Department of Energy under Contract No. DE AC02-05CH11231. We like to thank Roy Featherstone for his permission to base our work and his MATLAB code, and for answering our questions.

APPENDIX

The algorithms described here are part of the Computational Crystallography Open Source Toolbox (10), available under http://cctbx.sourceforge.net/. The source code for the procedures presented here can be found in these places:

- Rigidity analysis and tree generation with associated tests: scitbx/graph/
- TARDY model, Featherstone system model, joint library, tests using the reference potential object: scitbx/rigid_body/essence/
- Prototype code incl. scripts used to prepare Fig. 2: scitbx/rigid_body/proto/
- Scripts for running and evaluating tests using the real-space potential objects: mmtbx/refinement.
  The resulting plots are archived under:
  http://cci.lbl.gov/tardy_msndc2009/

Completely self-contained source code bundles are available under: http://cci.lbl.gov/cctbx_build/

REFERENCES

[1] http://www.rcsb.org/
[2] Carter, C.W. (ed.), 2003, Methods in Enzymology, Vol. 374, Academic Press
[3] Nocedal, J. and Wright S., 1999, Numerical Optimization, Springer
[4] Cate, J.H., Yusupov, M.M., Yusupova, G.Z., Earnest, T.N and Noller, H.F., 1999, Science, 285, 2095-2104
[5] Rice, L.M. and Brunger, A.T, 1994, Proteins 19, 277-290
[6] Adams, P.D., Grosse-Kunstleve, R.W., Hung, L.-W., Ioerger, T.R., McCoy, A.J., Moriarty, N.W., Read, R.J., Sacchettini, J.C., Sauter, N.K., Terwilliger, T.C., 2002, Acta Cryst., D58, 1948-1954
[7] Featherstone, R. 2007, Rigid Body Dynamics Algorithms, Springer
[8] Jacobs, D.J., 1998, J. Phys. A: Math Gen. 31, 6653-6668
[9] Jacobs, D.J. and Thorpe, M.F., 1998, US patent # 6014449.
[10] Grosse-Kunstleve, R.W., Sauter, N.K., Moriarty, N.W. and Adams, P.D., 2002, J. Appl. Cryst., 35, 126-136
[11] Mukherjee, R.M. and Anderson, K.S., 2007, Nonlinear Dyn., 48, 199-215
[12] http://axiom.anu.edu.au/~roy/spatial/
[13] Schwieters, C.D. and Clore, G.M., 2001, J. Mag. Res., 152, 288-302
[14] Jain, A., Vaidehi, N. and Rodriguez, G., 1993, J. Comp. Phys., 106, 258-268
[15] http://www.python.org/
[16] http://cctbx.sourceforge.net/scitbx_rigid_body_essence/
[17] Graver, J.E., 2001, Counting on Frameworks, Mathematical Association of America
[18] http://zinc.docking.org/

[19] http://www.ccdc.cam.ac.uk/
[20] Grosse-Kunstleve, R.W., Afonine P.V., Adams P.D., 2004, Newsletter of the IUCr Commission on Crystallographic Computing, 4, 19-36.
[21] Rossmann, M.G., Arnold E. (Ed.), 2001, International Tables for Crystallography, Volume F, Kluwer.
[22] Goldstein, H., Poole, C., Safko, J., 2002, Classical Mechanics, Addison Wesley.
[23] Afonine, P.V., Grosse-Kunstleve, R.W., Adams, P.D., Urzhumtsev, A., 2009, J. Appl. Cryst., in review.
[24] Giacovazzo, C. (Ed.), 1992, Fundamentals of Crystallography, Oxford University Press.
[25] Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S., Karplus, M., 1983, J. Comp. Chem., 4, 187-217.
[26] Hahn, T. (Ed.), 1996, International Tables for Crystallography, Volume A, Kluwer.