JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

Reciprocal Grids: A Hierarchical Algorithm for Computing Solution X-ray Scattering Curves from Supramolecular Complexes at High Resolution

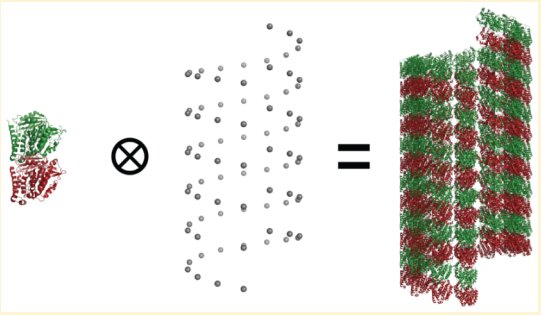
Avi Ginsburg,[†,‡] Tal Ben-Nun,[†,§] Roi Asor,[†] Asaf Shemesh,[†] Israel Ringel,[‡,∥] and Uri Raviv*[,†]

[†]The Institute of Chemistry and the Center for Nanoscience and Nanotechnology, [‡]The Institute for Drug Research, School of Pharmacy, and [§]Department of Computer Science, The Hebrew University of Jerusalem, Jerusalem 9190401, Israel

[∥]Azrieli College of Engineering, Jerusalem 9103501, Israel

Supporting Information

ABSTRACT: In many biochemical processes large biomolecular assemblies play important roles. X-ray scattering is a label-free bulk method that can probe the structure of large self-assembled complexes in solution. As we demonstrate in this paper, solution X-ray scattering can measure complex supramolecular assemblies at high sensitivity and resolution. At high resolution, however, data analysis of larger complexes is computationally demanding. We present an efficient method to compute the scattering curves from complex structures over a wide range of scattering angles. In our computational method, structures are defined as hierarchical trees in which repeating subunits are docked into their assembly symmetries, describing the manner subunits repeat in the structure (in other words, the locations and orientations of the repeating subunits). The amplitude of the assembly is calculated by computing the amplitudes of the basic subunits on 3*D* reciprocal-space grids, moving up in the hierarchy, calculating the grids of larger structures, and repeating this process for all the leaves and nodes of the tree. For very large structures, we developed a hybrid method that sums grids of smaller subunits in order to avoid numerical artifacts. We developed protocols for obtaining high-resolution solution X-ray scattering data from taxol-free microtubules at a wide range of scattering angles. We then validated our method by adequately modeling these high-resolution data. The higher speed and accuracy of our method, over existing methods, is demonstrated for smaller structures: short microtubule and tobacco mosaic virus. Our algorithm may be integrated into various structure prediction computational tools, simulations, and theoretical models, and provide means for testing their predicted structural model, by calculating the expected X-ray scattering curve and comparing with experimental data.



INTRODUCTION

Biomolecular assemblies are ubiquitous in nature and play many key roles in biological processes. Solution small/wide-angle X-ray scattering (SAXS/WAXS) methods allow for a thorough investigation of large self-assembled structures under various solution conditions (for example, pH, temperature, or osmotic pressure) in a noninvasive manner (label-free, for example). Various structural parameters can be determined, including characteristic distances, degree of folding, phase transitions, and their variation with time or conditions.[1−9] These bulk methods are highly sensitive to small structural changes, and even to the structure and density of the hydration shells around macromolecules.[10−12] Although SAXS and WAXS have recently become popular methods for obtaining structural data of biomolecules and nanomaterials, data interpretation remains challenging and is a bottleneck for their wider use in structural biology and biophysics, as well as in material science.[13,14] In this paper, we present a hierarchical algorithm for calculating solution X-ray scattering curves from large and complex structures made of many subunits, at high resolution, speed, and accuracy. As we demonstrate, our method provides opportunities to fully exploit the power of modern synchrotron solution X-ray scattering experiments.

Scattering data are measured as intensities (rather than amplitudes) in reciprocal $\vec{q}$-space. To elucidate structural parameters, models must be fitted to data. The modeling methods vary from coarse geometric models,[15,16] through a collection of spheroids,[9,17] to full atomic models.[8,12,18−20] Most methods (with the notable exception of the Debye formula,[21] which is only applicable for collections of isotropic objects), are approximations of the following solution scattering equations:

$$F(\vec{q}) = -r_0 \int \Delta\rho(\vec{r}) e^{i\vec{q}\cdot\vec{r}} d\vec{r} \tag{1}$$

$$I(q) = \langle I(\vec{q}) \rangle = \frac{1}{4\pi} \int_{\Omega_q} |F(\vec{q})|^2 d\Omega_q$$

$$= \frac{1}{4\pi} \int_0^{2\pi} d\phi_q \int_0^{\pi} |F(\vec{q})|^2 \sin(\theta_q) d\theta_q \tag{2}$$

Received: March 17, 2016
Published: July 13, 2016

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527

where $\vec{q}$ is the elastic scattering momentum transfer vector, $\vec{r}$ is the position vector in real-space, $\Delta\rho(\vec{r})$ is the electron density contrast with respect to the medium as a function of $\vec{r}$, $r_0 = 2.82 \times 10^{-5}$ Å is the Thomson scattering length, and F is the scattering amplitude. $I(q)$ is the orientation-averaged scattering intensity in solution as a function of the magnitude of the scattering vector, $q = |\vec{q}|$. $d\Omega_q = \sin(\theta_q)\, d\theta_q d\phi_q$ is the differential of the reciprocal-space solid-angle, where θ_q and ϕ_q are the polar- and azimuth-angles in $\vec{q}$-space, respectively. Equation 2 reduces the dimensions of the problem from three to one and describes structures in solution that are uniformly distributed in all orientations.

Use of low resolution geometric models to describe $\Delta\rho(\vec{r})$ is a rough approximation that is adequate at low scattering angles (see for example Supporting Information Figure S1).[15,22,23] Geometric models tend to have smooth surfaces and uniform layers or shells, and thus deviate from actual complex molecular architectures, which have fine-featured subunits, surfaces, shells, or layers. As geometric details are added, the resolution of models can increase and eventually approach atomic levels of detail. Atomic models are calculated by summing the contribution from the atomic form-factors according to their positions in real-space, which determine their phase difference.[7]

There are several approaches to compute the solution scattering intensity from a model containing n atoms. The Debye formula[21] computes the scattering contributions of all atom pairs in the structure, resulting in a computational complexity of $O(n^2)$. In practice, the Debye formula does not scale well with complex structures (large values of n); hence, various approximations are used. One approach[24] uses an octree data structure that scales with d_a^3, resulting in a complexity of $O((\mathrm{q}d_a)^3 \cdot n \log\left(\frac{n}{(\mathrm{q}d_a)^{1.5}}\right))$, where d_a is a "typical interatomic distance" of the scattering object. Another method[6,17] coarsens the computations to reduce the number of subunits, effectively reducing n. Alternatively, the Golden Vector (GV)[20] algorithm uses a quasi-random mesh to approximate orientation averaging. GV retains the accuracy of the Debye formalism with a computational complexity of $O(nw)$, where w is the complexity of the orientation averaging. A popular method, implemented in the CRYSOL[18] application, uses a spherical harmonics and Bessel function expansion of $e^{i\vec{q}\cdot\vec{r}}$. The infinite series is then truncated at l harmonics, leading to a computational complexity of $O(l^2\, n)$, where $l(2l + 1)$ is the number of terms of a spherical harmonics series (usually $l \ll n$). In all the above approximations, attaining a desired accuracy additionally depends on the structure geometry.

Both CRYSOL and GV deviate from the closed form Debye "gold standard" when calculating the scattering of larger non-spherical protein assemblies at higher resolutions ($q{\cdot}L \approx 100$, where L is the longest dimension, as shown later). The deviations are owing to truncation of the infinite spherical harmonics series in the case of CRYSOL, and numerical integration in the case of GV. These methods are meant for the computation of solution scattering intensities of smaller structures and do so accurately. It is only with the larger assemblies that deviations occur, owing to truncation of the spherical harmonics series and random sampling, respectively. Taking higher order harmonics into account can increase accuracy, but computation time increases quadratically with l.

Here we present an efficient and hierarchical method for computing the solution scattering intensities of large supramolecular assemblies. The approximation error of the method is constant with respect to the scattering angle, thereby enabling the simulation of large-scale structures at high resolution ($q{\cdot}L > 1000$).

To efficiently and accurately calculate the scattering intensities of large assemblies using our method, structures should be represented in a hierarchical manner (see Figure 1). The representation

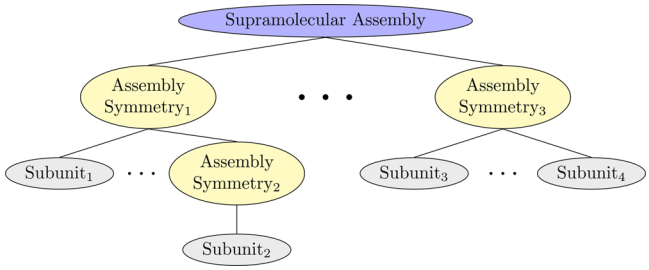


Figure 1. Example of modeling a supramolecular assembly in a hierarchical manner. Each *Assembly Symmetry* may contain multiple children. Children (of an *Assembly Symmetry*) can either be additional *Assembly Symmetries* or *Subunits*. A *Subunit* represents a geometric model or an atomic model, given as a Protein Data Bank (PDB) file. Internal nodes consist of *Assembly Symmetries*, whereas each leaf must be a *Subunit*. There may be an arbitrary number of hierarchy levels and nodes in each level. Many biological supramolecular self-assembled structures may be constructed that way. Care, however, should be taken when dealing with very large structures (see Results and Discussion for details).

differentiates between the unique basic elements (or subunits) and their assembly symmetries. A basic subunit can be a geometric model or an atomic model. The assembly symmetry describes the arrangement of identical subunits in a structure by providing the information about all the translations and rotations of a repeating subunit in the structure. This information can be given by defining a lattice or a list of locations and orientations for each repeating subunit. Figure 1 illustrates how an involved supramolecular assembly can be described in a hierarchical manner. At any level in the hierarchy, each model can be rotated, shifted, arranged in a lattice, or arranged in any other way in space. This process can then be repeated with other subunits or at higher levels in the hierarchy. Many complex structures can be expressed using this hierarchical representation, simplifying models to a small number of tree nodes (structural parameters), and thus their scattering curves can be readily computed.

For each unique subunit, we precompute the scattering amplitudes at each point on a predetermined 3D reciprocal $\vec{q}$-space grid (RG), tiling the relevant $\vec{q}$-space. To reduce the density of the RG, we use spline interpolations when evaluating the scattering amplitudes at intermediate $\vec{q}$ values between grid points. We then sum the contributions of all the subunits (identical or not) to the scattering amplitude according to their arrangement in real-space (see Materials and Methods Hierarchical Docking of Subunits). To further increase efficiency, all elements in the method are implemented in CUDA,[25] enabling the use of graphic processing unit (GPU) accelerators for a speedup of up to 3 orders of magnitude, depending on the complexity of the structure.

When possible, we compared our algorithm with the long-standing standard Debye, CRYSOL, and GV methods. When the structures were too large for these methods, we compared the models computed by our algorithm with high-resolution modern synchrotron solution X-ray scattering data. Using our algorithm, any level of structural complexity or granularity, including atomic resolution, can be evaluated in record time, outperforming existing methods. Our algorithm is particularly useful for modeling large structures with many repeating subunits.

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527

MATERIALS AND METHODS

The following method was developed to efficiently, accurately, and easily compute the solution X-ray scattering intensity from large supramolecular assemblies. Large assemblies ($n \gg 10^5$ atoms) comprised of many copies of smaller subunits are computationally expensive, are complicated, and hence, if possible, should be calculated using hierarchical methods.

Hierarchical Docking of Subunits. In real space, the electron density contrast of a complex structure made of repeating identical subunits is the convolution of the electron density contrast $\Delta\rho_j(\vec{r})$, describing the shape of object type j, with a collection of delta functions $\sum_k \delta(\vec{r} - \vec{r}_{j,k})$. The delta functions represent the shifts in real-space of these (identical) objects, where $\vec{r}_{j,k}$ is the shift of copy k of object type j. In reciprocal space, the convolution becomes a multiplication of the two contributions. The scattering amplitude $F_j(\vec{q})$ of object j is given by

$$F_j(\vec{q}) = -r_0 \int \Delta\rho_j(\vec{r}) \cdot e^{i\vec{q}\cdot\vec{r}} d\vec{r} \tag{3}$$

where eq 3 is the 3-dimensional Fourier transform of subunit of type j and $r_0 = 2.82 \times 10^{-5}$ Å is the Thomson scattering length, which we shall set to be 1. The lattice sum, or structure factor, is the Fourier transform of the sum of delta functions, $\sum_k \exp(i\vec{q}\cdot\vec{r}_{j,k})$, and is associated with the aforementioned relative shifts of the objects' centers of mass.

If copy k of object j is shifted by $\vec{r}_{j,m,k}$ and rotated by a rotation matrix $\mathbf{A}_{j,m}$ with respect to its principal axes, its electron density contrast is given by $\Delta\rho_j[\mathbf{A}_{j,m}^{-1}(\vec{r} - \vec{r}_{j,m,k})]$ and its Fourier transform is the scattering amplitude, given by[26] $F_j(\mathbf{A}_{j,m}^{-1}\vec{q}) \cdot \exp(i\vec{q}\cdot\vec{r}_{j,m,k})$.

Often, large assemblies can be described in a hierarchical manner, where subunits are shifted, rotated, and docked upon one another in all sorts of ways, which can be presented by a tree data structure (see Figure 1 and examples later in the text). The scattering amplitude of this type of large assemblies is

$$F(\vec{q}) = \sum_{j=1}^{J} \sum_{m=1}^{M_j} [F_j(\mathbf{A}_{j,m}^{-1}\vec{q}) \cdot \exp(i\vec{q}\cdot\vec{r}_{j,m})] \tag{4}$$

where $F(\vec{q})$ is the scattering amplitude from an assembly comprised of n_s subunits. J is the number of different types of objects, which is also the total number of leaves in the hierarchical tree structure representation of the entire supramolecular assembly (Figure 1). The leaves can either be geometry-based or atomic-based models, taken, for example, from Protein Data Bank (PDB) files. M_j is the number of instances (rotations and translations) of object type j, determined by rotation matrix $\mathbf{A}_{j,m}$ and real-space translation vector $\vec{r}_{j,m}$. The total number of subunits, n_s, is therefore $\sum_{j=1}^{J} M_j$.

Calculation of Atomic Models. The smallest unit usually considered is an ensemble of atoms (a protein subunit for example) positioned at specific locations, $\vec{r}_j$, in real-space. The scattering amplitude from atom j is calculated using the atomic form-factor five Gaussian approximation:

$$f_j^0(q) = \sum_{k=1}^{4} a_k \cdot \exp\left(-b_k\left(\frac{q}{4\pi}\right)^2\right) + c \tag{5}$$

where a_k, b_k, and c are the Cromer–Mann coefficients[27,28] that are given in units of r_0. The scattering amplitude of a molecule (whose structure can be taken from a PDB file, for example) is then given by[7]

$$F_{\text{Molecule}}(\vec{q}) = \sum_j f_j^0(q) \cdot \exp(i\vec{q}\cdot\vec{r}_j) \tag{6}$$

where $\vec{r}_j$ is the location of the jth atom in the molecule with respect to the origin. The random mean-squared displacement of each atom at finite temperatures, $\langle \vec{u}_{r_j}^2 \rangle$, can be included using the B-factors often provided in PDB files.[7] This contribution, however, was not included because there is another contribution that comes from the mean correlations, $\langle \vec{u}_{r_j}\vec{u}_{r_i} \rangle$, between the displacements of different atoms. The latter contribution opposes the former one and requires simulations or theoretical estimates, which are outside the scope of this paper.

Solvent subtraction can be applied in a plethora of methods.[4,5,11,12,18,29−34] One common method is to subtract Gaussian spheres localized at the center of each atom[30] and is the method employed in this work, when comparing to experimental data. The contrast is obtained by subtracting a "dummy" atom with a Gaussian electron density profile from each atom:

$$\rho_j^{\text{solvent}}(\vec{r}) = \rho_0 \exp[-(r/r_j^s)^2] \tag{7}$$

where ρ_0 is the mean electron density of the solvent ($\rho_0^{\text{water}} = 333$ e/nm^3), and r_j^s is the atomic radius of atom j in the PDB file.[18,30] For computation, published experimental r_j^s radii[30] were used and, where absent, replaced by empirical radii.[35] The scattering amplitude contribution of the Gaussian "dummy" atom is then given by

$$F_j^{\text{solvent}}(\vec{q}) = \int_0^{2\pi} d\phi_r \int_0^{\pi} d\theta_r \int_0^{\infty} \rho_0 e^{-(r/r_j^s)^2} \cdot e^{i\vec{q}\cdot\vec{r}} \cdot r^2 \sin\theta_r dr = \rho_0 \pi^{3/2} (r_j^s)^3 \exp[-(r_j^s \cdot q/2)^2] \tag{8}$$

where θ_r and ϕ_r are the polar and azimuth angles in real-space, respectively. The overall volume $V_{\text{ex}}^{\text{Gaussian}}$ is $\pi^{3/2}(r_j^s)^3$, and is larger by a factor of $\frac{3\sqrt{\pi}}{4} \approx 1.33$ than the volume of the corresponding uniform sphere, $V_{\text{ex}}^{\text{Uniform}} = \frac{4\pi}{3}(r_j^s)^3$, used elsewhere.[18] The value of the effective mean volume may be adjusted to some extent by fitting the value of ρ_0 in eq 8, which is in agreement with Fraser et al.[30] When the contribution of the solvent is calculated, the scattering amplitude from the molecule in the solvent is

$$F_{\text{Molecule in solvent}}(\vec{q}) = \sum_j [f_j^0(q) - F_j^{\text{solvent}}(q)] \cdot \exp(i\vec{q}\cdot\vec{r}_j) \tag{9}$$

Any other method[3−5,29] can easily be implemented as well.

Reciprocal Grid (RG) Method. Using geometric models[15,22,23] or atomic models, discussed in Calculation of Atomic Models, one can sample the reciprocal $\vec{q}$-space using a predetermined 3D mesh (or grid) of density d and calculate the scattering amplitude (from a specific object) for each $\vec{q}$ point in the grid. By storing and using this 3D reciprocal-space amplitude grid (RG) as a lookup table when calculating multiple copies and orientations of the same subunit, we avoid having to calculate the same contributions multiple times. Instead, for a real-space rotation $\mathbf{A}$ we search for the value of $F(\mathbf{A}^{-1}\vec{q})$ in the RG. After applying the rotation matrix, the magnitude of the vector $\mathbf{A}^{-1}\vec{q}$ is preserved, or $|\mathbf{A}^{-1}\vec{q}| = |\vec{q}|$. Its direction, however, changes to another value. This property makes polar coordinates a convenient choice for the RG geometry, as rotations of points do not incur interpolation along the q axis. For values between precomputed points in the RG, we use cubic

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527

spline interpolation to minimize interpolation errors. The density of the mesh should be at least enough to ensure five to six significant figures when interpolating; otherwise, the error of the calculated signals will become excessively large as computations move up in the hierarchy (see Application of the RG Method). The resulting RG is typically hundreds of megabytes to a couple of gigabytes in size. This approach was therefore infeasible when lower Random Access Memory (RAM) capacities were available.

Orientation Averaging. Once the final RG amplitudes representing the entire assembled structure are obtained, the total scattering amplitude is calculated at w solid angles. A squared norm operation is performed on the total amplitude at each solid angle, and the norms are then averaged. For example, using Monte Carlo for averaging would evaluate the following scattering intensity:

$$I(q) = \frac{\pi}{2w} \sum_{i=1}^{w} |F(\vec{q}_i)|^2 \sin(\theta_q^i) \tag{10}$$

where $\vec{q}_i = (q, \theta_q^i, \phi_q^i)$.

The range of choices for numerical integration methods varies greatly, each with its advantages and disadvantages. Adaptive methods (for example, Gauss−Kronrod[36]) are difficult to implement on a GPU (partly owing to their inherent recursive character), whereas nonadaptive ones can be wasteful in terms of calculations. We found that using the VEGAS Monte Carlo method[37] gives many of the advantages of both approaches, especially when dealing with long objects, whose Fourier representation is highly oscillatory. This property stems from the fact that VEGAS attempts to minimize the variation in the sampling variance.

Application of the RG Method. Using this form, the scattering intensity can be calculated (with a few exceptions, noted later) as follows.

1. Calculate the complex scattering amplitude of the smallest subunit (leaf in the tree structure) at each point on the $\vec{q}$-space grid (RG) and save it in memory.
2. Move up the hierarchy (see Figure 1) and calculate the RG amplitudes in $\vec{q}$-space of the larger unit, using interpolations from the precalculated RG amplitudes of the smaller subunits or assemblies. When necessary, save the new RG amplitude in memory and discard (or cache) the old.
3. Repeat previous step until the RG amplitude of the full assembly is created.
4. Numerically approximate the orientation-averaged intensity using the final RG amplitude.

The inherent advantages of this method are several. First, speed and accuracy are retained even when the size of the structure is very large and the shape is nonspherical. The same levels of accuracy and speed cannot be attained by using truncated series of spherical harmonics,[18] which would be inefficient and often inaccurate (at higher q-values in particular, as demonstrated later). Second, subunits are allowed to be both translated and rotated arbitrarily, and not simply translated at the same orientation on a lattice. This capability is useful for modeling assemblies that contain copies of the same subunit at different orientations (for example, a helical lattice). Third, once RG amplitudes are calculated, changes in positions and orientations of the subunits only require interpolation operations. Finally, the algorithm has inherent data parallelism (in computing each point on the grid) as well as task parallelism (computing different segments of the tree in parallel).

These advantages are achieved because the translation/rotation calculations are done in Fourier ($\vec{q}$) space (other programs, for example, SASREF,[19] also use rotations in Fourier space, but are limited to specific symmetries). The computation process takes advantage of the properties of Fourier transforms (convolution, linearity, position-shifting, and rotation theorem), which allow multiple copies of a single subunit to be calculated by multiplying the Fourier transform of a subunit by the Fourier transform of the sum of delta functions, located at the centers of mass of the other copies, $\vec{r}_{j,m,k}$. Rotation of a subunit in real-space by $\mathbf{A}_{j,m}$ is equivalent to the same rotation in reciprocal space. The RG is then used to retrieve the corresponding scattering amplitude at the rotated $\vec{q}$ vector, $\mathbf{A}_{j,m}^{-1}\vec{q}$. Additionally, owing to the computational concurrency, implementation on GPUs is both possible and desirable. When developing the underlying algorithms, targeting massively parallel architectures has affected some of our implementation choices, for example, creation of efficient data structures for optimal memory access bandwidth. Note that the RG method can also be efficiently implemented for other computing architectures (e.g., multicore CPUs). To conclude, the RG method can scale from a single CPU core to a supercomputer with many GPUs, each consisting of thousands of cores.

Computational Complexity of the RG Method. If the structure of interest is assembled from multiple repeating subunits that may have different amplitudes (owing to their different orientation or structure), we can reduce the computation time by computing an RG of size G (determined by the grid density times the grid volume). Assume that this structure is described by a tree data structure (Figure 1), where each leaf describes a subunit with p_i atoms. To compute the RG amplitude of the ith leaf, the contributions from all the p_i atoms must be summed at each point in the reciprocal-space grid using eq 6 or 9 (for vacuum or solution). The resulting computational complexity is $O(G{\cdot}p_i)$.

Internal nodes of the hierarchy tree are assembly symmetries with a list of S instances of its children (at different translations and orientations). The computation of an internal node consists of a sum of its instances' contributions. Thus, if there are C children, the complexity is $O(G{\cdot}S{\cdot}C)$ (see Figure 2).

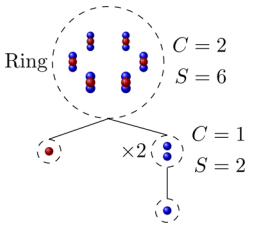


Figure 2. Example of a hierarchy that describes the variables used to define trees. In this abstract structure a six point symmetry (S) forms a ring from two children (C), a red sphere and a pair of blue spheres. The pair of blue spheres is itself a node comprised of the same blue sphere at two different positions.

This complexity can be lowered to $O(G{\cdot}S{\cdot}C)$ if the children's contributions are first summed, and then the instances are computed. The difference, however, is negligible, as C is usually very small (often $C = 1$).

By using a series of RGs to construct the scattering amplitude of an entire tree (one for each node in the hierarchy, see Figure 1), the computational complexity for generating the root grid can be reduced to

$$O\left(G{\cdot}\left(\sum_{i=1}^{J} p_i + \sum_{j=1}^{y} S_j{\cdot}C_j\right)\right) \tag{11}$$

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527

Here, y is the number of internal nodes (assembly symmetries) and J is the total number of leaves. The number of atoms in this structure is $n = \sum_i^J p_i \cdot \prod_{j \in T_i} S_j = \sum_i^J n_i$, where T_i is the group of nodes above the ith leaf. This translates to a complexity that is sublinear in n for the amplitude computation, because n is a sum of products $(\sum_i^J p_i \cdot \prod_{j \in T_i} S_j)$ and the complexity is a summation of the same elements $(\sum_{i=1}^J p_i + \sum_{j=1}^y S_j \cdot C_j)$.

For the intensity computation, an additional orientation average (numerical integration) is required, as in eq 10. For m points along the q-axis, w points need to be evaluated from the structure's total amplitude, contributing $O(m \cdot w)$. Note that w, which governs the accuracy of the resulting intensity curve, depends on the integration algorithm, sampling method, and distribution.

On a practical note, precomputed RG amplitudes of subunits may be loaded and used to calculate many different structures containing the same subunits. The computation time is then proportional to the evaluation of internal nodes and orientation average, hence considerably shorter.

Grid Density. As indicated above, for large self-assembled structures (for example, subunits arranged along a long helix), the scattering amplitude becomes highly oscillatory in $\vec{q}$-space. If the largest distance within the object is L, the scattering amplitude will oscillate approximately as $\sin\left(q \cdot \frac{L}{2}\right)$. According to the Nyquist–Shannon sampling rate,[38,39] accurate representation using an RG requires at least two points per period. Hence, the grid density should be at least $(\Delta q)^{-3} \approx (L/2\pi)^3$ and the grid size should be at least $G \approx \left(q_{\max} \cdot \frac{L}{2\pi}\right)^3$, where $q_{\max}$ is the maximum q-range. For atomic models, the scaling of G may vary from n (spherical shapes, best case) to n^3 (elongated, worst case). Specifically in the implementation used for this paper we used a single grid density for all levels in the hierarchy. In order to increase grid density, for all points in the higher density the value would be interpolated from the previous density.

In the case of large structures, the RG method could require an incapacitatingly dense mesh for current RAM sizes and computation times. Below, we present two ways to overcome this hindrance.

Direct Computation. In the direct method no RG amplitudes are computed. Instead, each geometry or PDB object is identified, and all its copies (orientation and location) in the structure are collected. To calculate the intensity, scattering amplitudes at various $\vec{q}$ are evaluated and squared for orientation averaging (eq 10). The scattering amplitudes are directly computed by modifying eq 4 so that each node in the tree structure is grouped by its orientation. If the same orientation (j,m) is observed in repeating subunits in the structure, the contribution of that subunit to the total scattering amplitude is reused, with the relevant phase contribution associated with its real-space locations, given by the vectors $\vec{r}_{j,m,k}$:

$$F(\vec{q}) = \sum_{j=1}^{J} \sum_{m=1}^{M_j^u} \left[F_j(\mathbf{A}_{j,m}^{-1}\vec{q}) \cdot \sum_{k=1}^{K_{j,m}} \exp(i\vec{q} \cdot \vec{r}_{j,m,k})\right] \tag{12}$$

J is the number of different types of objects (leaves in the hierarchical tree, see Figure 1). M_j^u is the number of unique orientation of object type j, determined by the rotation matrices $\mathbf{A}_{j,m}$. $K_{j,m}$ is the number of real-space translations of object j in orientation $\mathbf{A}_{j,m}$.

The total number of subunits n_s is $\sum_{j=1}^J \sum_{m=1}^{M_j^u} K_{j,m}$. The complexity of this calculation is $O(w \cdot \sum_{j=1}^J M_j^u \cdot p_j)$, where p_j is the number of atoms of subunit j (note that $n \geq \sum_{j=1}^J M_j^u \cdot p_j$). This method sidesteps undersampling of the RG (below the Nyquist rate), as it avoids the interpolations inherent in the RG method at the cost of many more amplitude evaluations.

RG/Direct Hybrid Computation. A better solution for the case of large structures is to combine the direct computation with the RG method. In this hybrid method, RG amplitudes are calculated from the leaves up to predetermined nodes in the assembly hierarchy tree structure (see Hierarchical Docking of Subunits, Application of the RG Method, and Figure 1), and from these nodes upward computed as subunits in the direct method. A node's grid density is inversely proportional to its structures largest dimension cubed. Therefore, the last node's grid density can be significantly lower than required of the larger structure.

Differently oriented subunits are directly calculated as in the direct method; however, the highest calculated RG amplitudes are accessed as $F(\mathbf{A}^{-1}\vec{q})$, as explained in Reciprocal Grid (RG) Method. In the hybrid method, at least one of the leaves in the hierarchical tree structure (a geometric or a PDB model) is calculated to an RG. RG amplitudes may be calculated for internal nodes, and if they are, then their leaves are discarded and the internal nodes are treated as leaves.

The hybrid method averts the use of RGs that represent long structures, thus providing a solution with arbitrary accuracy in the face of the RG memory/accuracy trade-off. This approach is effectively a coarse-grained method with near-atomic resolution accuracy, where the subunit count (n) can be much smaller than the number of atoms (see Supporting Information Figure S2). When only leaves are computed to an RG, the resulting computation complexity is then

$$O\left(G \cdot \sum_{i=1}^{J} p_i + m \cdot w \sum_{i=1}^{J} \left(\frac{n_i}{p_i}\right)\right) \tag{13}$$

where $\frac{n_i}{p_i}$ is the total number of instances of subunit i, or $\prod_{j \in T_i} S_j$. The advantage of the hybrid method is that G is dependent on the length of the subunit rather than the length of the entire structure.

Computer Architectures Used. The RG method and CRYSOL v2.8.3 were run on a Windows computer with an Intel Core i5 3.2 GHz CPU, while the Golden Vector (GV) calculations were run under Linux (Ubuntu distribution) with the same processor. The GPU versions were run on an NVIDIA Titan GPU.

Experimental Section. *Tubulin Purification.* Tubulin was purified from porcine brains by three cycles of polymerization and depolymerizations. The first cycle was done at low salt, as described,[40,41] to allow assembly in the presence of microtubule associated proteins, which reduce the critical assembly concentration of tubulin. This reduction enabled higher yield, as the first cycle is performed with high solution volumes. The other two cycles were performed in a high-molarity buffer, as described.[42]

X-ray Scattering Measurements. Solution X-ray scattering measurements were performed in the ID02 beamline at the ESRF synchrotron (Grenoble, France). The beam size was 400 × 200 μm^2 (horizontal and vertical fwhm, respectively) at the sample position. A Fast-Readout Low-Noise (FReLoN) Kodak KAF-4320 image CCD based sensor was used (area 100 × 100 mm^2). The energy of the X-ray photons was 10 keV. A 2 mm quartz capillary flow-cell was used for the measurements.

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527

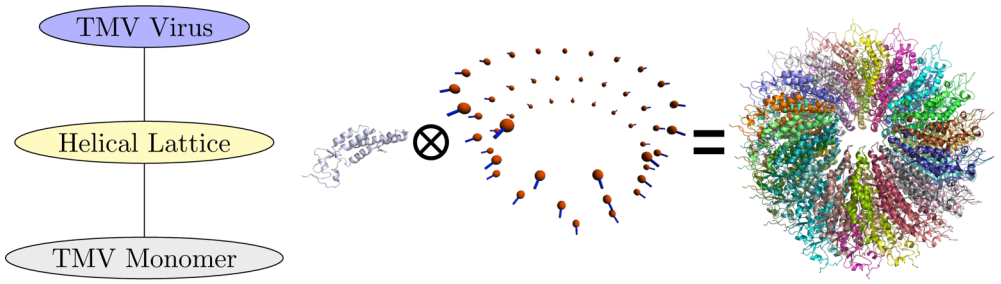


Figure 3. Construction of the TMV capsid in a hierarchical manner. The assembly of the TMV can be constructed from the capsid protein monomer by convolving it with a helical lattice. It is critical to note that the convolution is not with simple delta functions, as that would result in all the monomers having the same orientation. Rather, the construction must incorporate the rotation matrix of each individual copy (eq 4), so that all the monomers point outward in the direction of the blue lines. The hierarchical tree data structure of the TMV is shown on the left.

20 mg/mL tubulin in PEM buffer (80 mM 1,4-piperazinediethanesulfonic acid, 1 mM $MgCl_2$, 1 mM EGTA, adjusted to pH = 6.9 with KOH) supplemented with 4 mM guanosine-5′-triphosphate (GTP), were incubated at 25 °C for 30 min. The polymerized tubulin solution was then measured in the flow cell capillary. As small tubulin assemblies and residual dimeric tubulin coexist with microtubules, the sample was centrifuged at 20,800 g, at 25 °C for 30 min and resulted in a microtubule pellet. The supernatant, which contained the coexisting small tubulin assemblies and dimeric tubulin (up to its critical concentration), were measured at the same spot in the flow-cell capillary and the resulting scattering curve served as background (see Supporting Information Figure S3).

RESULTS AND DISCUSSION

Herein, we demonstrate our methods for three examples and, when possible, compare the accuracy and computation time with existing methods. For larger structures we compare our method with high-resolution synchrotron solution X-ray scattering data. As examples, we use the tobacco mosaic virus (TMV) and short- and long-microtubules (MT).

Tobacco Mosaic Virus. TMV is a rod-shaped plant virus with a typical length of 300 nm, a diameter of 18 nm, and a central channel of approximately 4 nm. The virus contains RNA and the coat protein (158 amino acids, ∼17 500 Da) subunits that assemble into a right-handed helix. Three complete turns contain 49 subunits.[43] The PDB file 2OM3 is comprised of 49 copies of the capsid protein monomer together with the piece of RNA that binds to the protein (1279 atoms per monomer, including the RNA) and forms a cylinder that is ∼6.9 nm long. The entire complex is made from exact copies of the monomer, as can be confirmed by aligning any monomer to the other 48.

The monomers are arranged in a right-handed helical symmetry, and the helical rise per subunit, along the long, z, axis is $\frac{6.9}{49}$ nm. Likewise, neighboring monomers are rotated around the z axis by $2\pi\cdot\frac{3}{49}$ in relation to each other. For a longer cylinder, the 49-mer could be copied along the z-axis with a displacement of 6.9 nm.

To calculate the scattering amplitude of TMV using the RG method (eq 4 and the Reciprocal Grid (RG) Method), one would compute the RG amplitude of the monomer (1279 atomic form factor evaluations per grid cell), plus 49 evaluations for the different translations/rotations of the monomer subunit. To make the TMV model 10 times longer, another 10 evaluations for the translations of the 49-mer would be required. Instead of $1279 \times 49 \times 10$ evaluations if the amplitude is calculated directly, using an RG with G cells would only entail $G\cdot(1279 + 49 + 10)$ computations. This reduction in computational complexity applies only when the object is constructed in a hierarchical fashion (Figure 3). The scattering intensity is then obtained by squaring the amplitudes at w different orientations (eq 10).

Comparison with Existing Methods. To demonstrate the effectiveness of the RG method, we compared its computed scattering signal for the *in vacuo* TMV structure with the scattering signals calculated by the golden vector (GV) method and CRYSOL. The ground-truth for comparison was created using the closed-form Debye scattering formula (with computational complexity of $O(n^2)$). Both the subunit and the biological assembly (the 49 subunits) were created and compared with the Debye version (Figures 4 and 5, respectively).

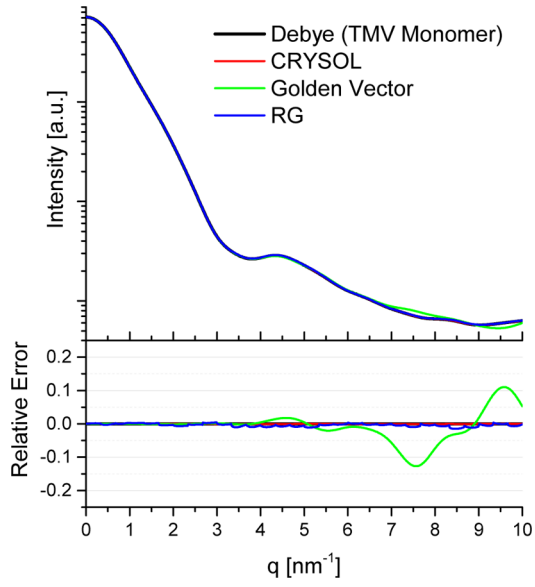


Figure 4. Calculated scattering intensity of the TMV capsid protein monomer using Debye, CRYSOL, golden vector (GV), and our RG methods, as indicated in the figure. All the methods give roughly the same scattering curve (top). The CRYSOL curve was generated with maximal accuracy settings (degree of harmonics = 50). For the golden vector method, 201 vectors were used. The relative errors were calculated against the Debye calculation (bottom). The maximal error for the RG method is ≈1.5% (at q = 8.46 nm^{-1}). Table 1 compares this value with the maximal errors of the other methods.

In the case of the subunit (Figure 4), the three methods successfully created the same scattering signal as the Debye method. Table 1 shows the calculation times and maximum errors of each method. Based on accuracy and calculation times on a CPU, CRYSOL is the preferred method for the subunit. When looking at the assembly (Figure 5), however, the table

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527

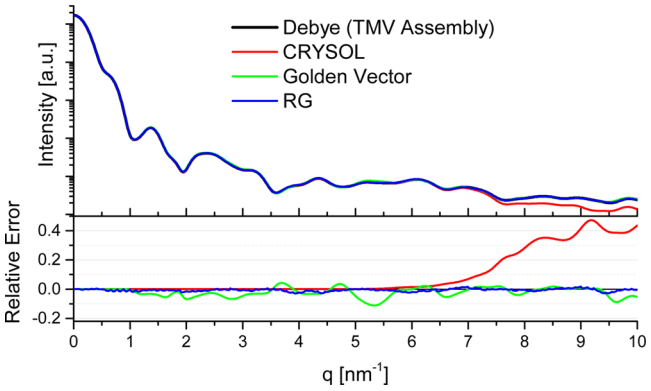


Figure 5. Calculated scattering curves of the entire TMV assembly (PDB file 2OM3, see Results and Discussion). The general shapes of the calculated signals coincide, with the notable exception of CRYSOL at $q > 7$ nm^{-1} (top). We used the same generation parameters as in Figure 4. The relative errors shown at the bottom were calculated against the Debye calculation (black curve).

turns both in terms of accuracy and calculation times. CRYSOL is accurate at $q < 7$ nm^{-1}, but then a systematic deviation, as a result of the truncation, is observed. The golden vector method deviates from the Debye method by up to 10%, the same as with the monomer. The CRYSOL and golden vector methods scale linearly with n and take a little more than the expected 49 times the subunit calculation time. On the other hand, the RG method has a maximal error of ≈3.6% whereas calculation times are about 13 times longer than the monomer and, hence, scale sublinearly with the number of atoms.

Short Microtubule. When computing larger structures, like microtubules (MTs), the accuracy of the RG method stands out even more. A model MT was constructed using a helical lattice where the pitch (12.195 nm) and radius (11.940 nm) were derived from the PDB file 3J6F.[44] The lattice model was a three start left-handed helix with 14 protofilaments, of 12 tubulin dimers each (Supporting Information Figure S6). The pitch and radius refer to the tubulin dimer subunits' centers of mass (rather than the inner or outer radii). The calculated *in vacuo* SAXS curves and deviations from the Debye method are shown in Figure 6, presenting the advantage of our method, as was shown for TMV. Calculation times and errors are shown in Table 1. The time ratio for the CRYSOL methods is roughly twice the expected 168 it would be if linearly scaled with the number of atoms. Using the RG method, with a grid density adapted for the size of the object, the ratio is significantly less than the number of subunits. The error in the RG method oscillates around zero, whereas the CRYSOL error is systematic from $q = 1$ nm^{-1} and up, causing the high maximal error.

Long Microtubule: Comparison with High-Resolution Scattering Data. The importance of modeling protein structures at atomic resolutions is demonstrated by comparing high-resolution third generation synchrotron solution X-ray scattering data from taxol-free MTs along with an atomic model of MT in solution. A Gaussian sphere representing the solvent is subtracted from each atom in the model (eq 9).

In earlier studies,[45,46] taxol-stabilized MT SAXS curves presented data below $q = 2$ nm^{-1}. We developed experimental protocols (see the Experimental Section and Supporting Information Figure S3) to obtain high-quality scattering data from taxol-free MTs beyond $q = 7$ nm^{-1} with distinguishable features throughout (Figure 7). Our model adequately fits the data. At high resolutions, seemingly small differences can incur a significant change in the scattering signal. For example, changing the PDB file from 3J6F to 1JFF leads to noticeable changes in the modeled scattering curves (Supporting Information Figure S7) even though the atom position root-mean-square-deviation (RMSD) between the PDB files is about 1.1 Å.

When attempting to calculate a much shorter MT (containing 168 dimers) with CRYSOL, the results significantly deviated from the RG and Debye methods, shown in Figure 6. The error in CRYSOL is systematic and determined by the upper limit on the spherical harmonic expansion, whereas the error in the RG method is random and determined by the orientation sampling. The CRYSOL calculation times were about 8 h for an MT with a 12-dimer height, as opposed to about 60 s or less on the GPU for the RG method (Table 1). Although this comparison lacks an exact common denominator, the RG method is more accurate and faster. Both the CPU and the GPU used the hybrid method (see RG/Direct Hybrid Computation) and returned similar results. The hybrid method was used because creating a single RG amplitude of the entire MT structure resulted in numerical artifacts, as presented in Supporting Information Figure S2.

Table 1. Calculation Times and Maximal Error for the Generated Models of TMV and Short Microtubules in Each Method[a]

	Method	Subunit [s]	RMS/Max Error [%]	Assembly [s]	RMS/Max[b] Error [%]	Ratio
TMV	CRYSOL	6	0.05/0.10	1045	18.1/47.2	174
	Golden Vector	63	4.5/12.7	3184	3.8/11.2	50.54
	RG CPU[c]	18.2 (0.333)	0.58/1.5	246.2 (0.125)	1.22/3.6	13.5
	RG GPU	2.21 (0.333)	0.50/0.95	6.06 (0.125)	1.14/3.5	2.74
	Debye CPU	25.64	-	55400	-	2160
	Debye GPU	0.382	-	169.2	-	442.9
Tubulin[d]/Short MT	CRYSOL	50	0.28/0.63	26828	77.3/97.6	537
	RG CPU	122.3 (0.25)	0.91/2.1	4493.6 (0.25)	1.38/4.3	36.7
	RG GPU	3.18 (0.25)	0.82/1.8	59.11 (0.25)	0.9/3.3	18.6
	Debye CPU	640	-	5.2×10^6	-	8125
	Debye GPU	5.4	-	5.7×10^4	-	1.1×10^4

[a]See Computer Architectures Used and Figure 4 for the generation parameters. For the RG method, VEGAS integration was used for orientation averaging on the GPU, and Monte Carlo on the CPU. The RG step size (in units of nm^{-1}) is indicated in parentheses next to the computation time. Both the root mean squared (RMS) and the maximum error from the Debye method are listed. [b]The errors of the RG method oscillate around zero, whereas the errors of the other methods are more systematic when calculating the assemblies. [c]The RG method was mostly parallelized in our implementation, and the time listed mostly consists of CPU overhead. The CPU times displayed in the table correspond to a single core CPU (or single-threaded implementation). [d]See Supporting Information Figure S5.

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527

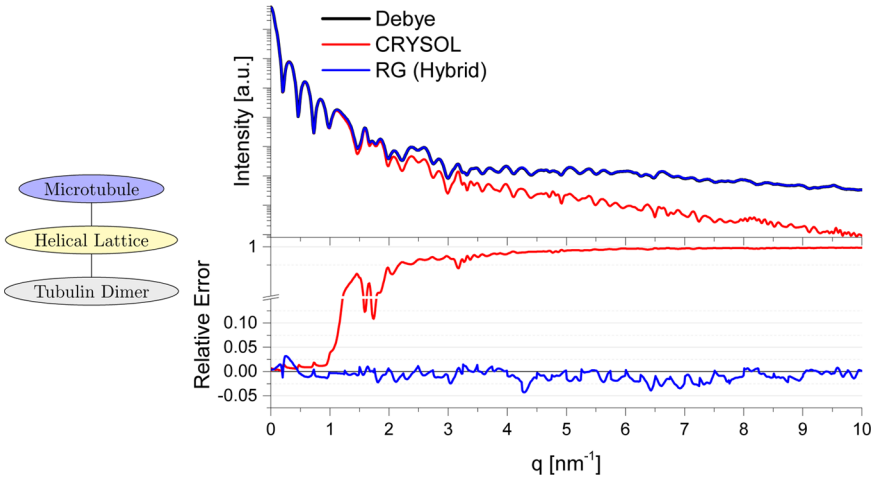


Figure 6. Calculated scattering curves of a short microtubule (168 tubulin dimers, 14 protofilaments of 12 dimers each). The RG method (blue curve) retains a small error throughout the entire q-range, whereas the CRYSOL curve (red curve), obtained by using the generation parameters listed in Figure 4, deviates significantly from $q > 1$ nm^{-1}. The RG curve was generated using the hybrid method (see RG/Direct Hybrid Computation). The Fourier space representation of the tubulin dimer was calculated and saved as an RG amplitude. Consequent computations required for the orientation averaging of the 168 positions and orientations were accessed from the single RG amplitude (with a density of $\approx(8\text{ nm})^3$ as opposed to creating an RG that accounts for the entire structure (estimated density of $\approx(100\text{ nm})^3$, which is 12^3 times denser). The diagram on the left shows the tree representation used by the hybrid method. The relative errors at the bottom were calculated against the Debye computation (black curve). A version of CRYSOL able to compute up to a harmonics degree of 100 was provided by Dmitri Svergun. Higher degree harmonics somewhat increase accuracy, as shown in Supporting Information Figure S4.

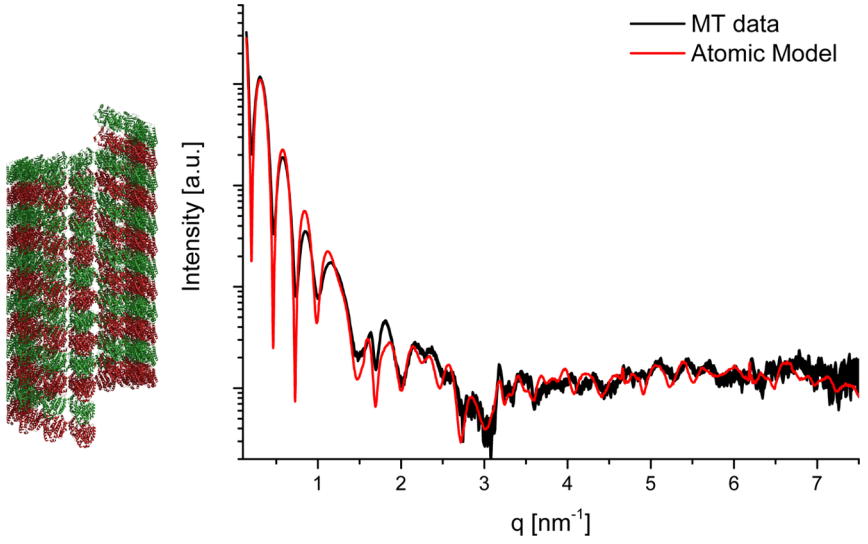


Figure 7. Taxol-free microtubule (MT) scattering data and an atomic model. Twenty mg/mL tubulin was assembled into MTs by adding 4 mM guanosine-5′-triphosphate (GTP) and incubating the solution for 30 min at 25 °C. The scattering from the MT solution was measured. MTs were then spun down, and the scattering curve from the supernatant was measured and used as background (the supernatant contained the buffer and the coexisting tubulin dimers and small tubulin assemblies). The scattering intensities were azimuthally integrated. The black curve shows the background-subtracted solution X-ray scattering intensity (see Supporting Information Figure S3 and Experimental Section). The black line thickness corresponds to the experimental error of the data. The resulting signal has distinct features in the entire measured range 0.1 nm^{-1} < q < 7.5 nm^{-1}. Using structural parameters from Short Microtubule but with 48 (instead of 12) dimers along the long MT axis, and the hybrid method, a theoretical intensity curve was computed in about half a minute on an NVIDIA Titan GPU (red curve). The hybrid method (see RG/Direct Hybrid Computation) was used because when the structure is that long, multiple layers of hierarchy may lead to artifacts (Supporting Information Figure S2). The diagram on the left shows a segment of the computed MT atomic model.

Supporting Information Figure S1 compares the atomic model with a uniform hollow cylinder model and another geometric model, which uses a sphere with a 2 nm radius to represent each tubulin monomer (two spheres per $\alpha\beta$ dimer). In the latter geometric model, these spheres were placed on the same helical lattice as the atomic model. Supporting Information Figure S1 shows that atomic models can provide a superior fit to the data at a wide q-range.

While our method creates a wide range of opportunities in modeling solution scattering curves, there are a few limitations that need to be addressed. Using RGs for SAXS has the same drawbacks of using lookup tables in general. The amount of memory required for our 3D RG scales like the cubed density on the q-axis. Thus, if the Fourier space oscillates rapidly (as it does for large or long objects), then the required RG size increases even more. The size increase leads to a cap on the size of a

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527

structure, above which the only accurate method is the hybrid method (see Direct and Hybrid Methods). Using RGs on large objects can result in artifacts, as shown in Supporting Information Figure S2. If, however, high accuracy is not required, or the structure is not elongated (yet may possibly contain many subunits), a sparser RG can be used, speeding up the computations significantly.

Any solvent subtraction method[5,29,30] could easily be integrated into the RG method (see Calculation of Atomic Models). Caution, however, should be used if employing calculation of solvation layers in conjunction with large protein assemblies. When positioning multiple proteins in proximity, their solvation layers may overlap, causing excess scattering. Moreover, it is reasonable to assume that the solvation layer will differ, as the neighboring proteins were presumably not taken into account during the initial calculation.

The RG method can also be used to combine geometric and atomic models (for example, placing an atomic modeled protein next to a lipid membrane that was modeled by a geometric multilayer structure). Nevertheless, geometric models that are practically infinite in at least one dimension[15,22,23] lead to a near delta function in the Fourier space, which is impossible to model accurately using an RG. Geometric models should therefore be limited to a finite size (up to a few hundreds of nanometers). Thus, caution should also be employed when combining long or very large geometric models.

CONCLUSIONS

We have presented a method to calculate solution X-ray scattering curves of large supramolecular (in particular protein) assemblies at high resolution, speed, and accuracy. As opposed to several other existing methods[15,22,23] that allow accelerated calculation of identical subunits only on a lattice, our method allows the subunits to be arbitrarily rotated, while retaining the speedup. Our method creates a vast range of opportunities in advanced and high-resolution solution scattering data analysis. Owing to the relatively low evaluation cost of our method, curve fitting algorithms can now be readily applied to reveal structural parameters of large supramolecular assemblies containing rigid monomers. In the future, it should be possible to incorporate our method into computer simulations and structure modeling and prediction computational tools[47] so that the solution scattering curves of the predicted structures will be evaluated and compared with measurements. This incorporation will potentially help to better model solution X-ray scattering measurements, on the one hand, and provide experimental feedback to structural biology and biophysics computational tools and computer simulations, on the other. Solution X-ray scattering is a label-free bulk method. The measurements are highly sensitive to small structural changes; hence, the advancement of our methods addresses an urgent need in structural biology and biophysics as well as in material science.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.6b00159.

Atomic (high-resolution) and geometric (low-resolution) models; numerical artifacts owing to inadequate use of the RG method; solution SAXS from the MT solution; effect of spherical harmonics order; calculated scattering intensity of the tubulin dimer; microtubule models; and computed microtubule (MT) scattering curves using different tubulin PDB files (PDF)

AUTHOR INFORMATION

Corresponding Author

*E-mail: uri.raviv@mail.huji.ac.il. Phone: +972 (2) 6586030. Fax: +972 (2) 5660425.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We are grateful to Daniel Harries for helpful discussions. ESRF, beamline ID02 (T. Naryanan and his team), is acknowledged, as the data presented in the paper were acquired there. We thank Dmitri Svergun for providing us with a special version of CRYSOL. AG, RA, and AS and this project were funded by Israel Science Foundation (1372/13), US-Israel Binational Science Foundation (2009271), Israel Ministry of Science, and Focal Technology Area—Hybrid Nanomaterials program of the Planning and Budgeting Committee of the Israel Council of Higher Education. We thank the Safra, Wolfson, and Rudin Foundations for supporting our laboratory.

NOMENCLATURE

RG, reciprocal grid; MT, microtubule; TMV, tobacco mosaic virus; SAXS, small-angle X-ray scattering; WAXS, wide-angle X-ray scattering; GPU, graphic processing unit; PDB, protein data bank; GV, Golden Vector

REFERENCES

(1) Ojeda-Lopez, M. A.; Needleman, D. J.; Song, C.; Ginsburg, A.; Kohl, P. A.; Li, Y.; Miller, H. P.; Wilson, L.; Raviv, U.; Choi, M. C.; Safinya, C. R. *Nat. Mater.* **2014**, *13*, 195−203.
(2) Kler, S.; Asor, R.; Li, C.; Ginsburg, A.; Harries, D.; Oppenheim, A.; Zlotnick, A.; Raviv, U. *J. Am. Chem. Soc.* **2012**, *134*, 8823−8830.
(3) Sorenson, J. M.; Hura, G.; Glaeser, R. M.; Head-Gordon, T. *J. Chem. Phys.* **2000**, *113*, 9149−9161.
(4) Bardhan, J.; Park, S.; Makowski, L. *J. Appl. Crystallogr.* **2009**, *42*, 932−943.
(5) Park, S.; Bardhan, J. P.; Roux, B.; Makowski, L. *J. Chem. Phys.* **2009**, *130*, 134114.
(6) Chacon, P.; Moran, F.; Diaz, J.; Pantos, E.; Andreu, J. *Biophys. J.* **1998**, *74*, 2760−2775.
(7) Als-Nielsen, J.; McMorrow, D. *Elements of modern x-ray physics*; John Wiley & Sons: 2011.
(8) Ravikumar, K. M.; Huang, W.; Yang, S. *J. Chem. Phys.* **2013**, *138*, 024112.
(9) Svergun, D. *Biophys. J.* **1999**, *76*, 2879−2886.
(10) Zheng, W.; Tekpinar, M. *Biophys. J.* **2011**, *101*, 2981−2991.
(11) Poitevin, F.; Orland, H.; Doniach, S.; Koehl, P.; Delarue, M. *Nucleic Acids Res.* **2011**, *39*, W184−W189.
(12) Schneidman-Duhovny, D.; Hammel, M.; Tainer, J. A.; Sali, A. *Biophys. J.* **2013**, *105*, 962−974.
(13) Franke, D.; Jeffries, C. M.; Svergun, D. I. *Nat. Methods* **2015**, *12*, 419.
(14) Rambo, R. P.; Tainer, J. A. *Nature* **2013**, *496*, 477−481.
(15) Székely, P.; Ginsburg, A.; Ben-Nun, T.; Raviv, U. *Langmuir* **2010**, *26*, 13110−13129.
(16) Ilavsky, J.; Jemian, P. R. *J. Appl. Crystallogr.* **2009**, *42*, 347−353.
(17) Svergun, D. I.; Petoukhov, M. V.; Koch, M. H. *Biophys. J.* **2001**, *80*, 2946−2953.
(18) Svergun, D.; Barberato, C.; Koch, M. *J. Appl. Crystallogr.* **1995**, *28*, 768−773.
(19) Petoukhov, M. V.; Svergun, D. I. *Biophys. J.* **2005**, *89*, 1237−1250.
(20) Watson, M. C.; Curtis, J. E. *J. Appl. Crystallogr.* **2013**, *46*, 1171−1177.
(21) Debye, P. *Ann. Phys.* **1915**, *351*, 809−823.

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527

(22) Ben-Nun, T.; Ginsburg, A.; Székely, P.; Raviv, U. *J. Appl. Crystallogr.* **2010**, *43*, 1522−1531.
(23) Ben-Nun, T.; Asor, R.; Ginsburg, A.; Raviv, U. *Isr. J. Chem.* **2015**, DOI: 10.1002/ijch.201500037.
(24) Gumerov, N. A.; Berlin, K.; Fushman, D.; Duraiswami, R. *J. Comput. Chem.* **2012**, *33*, 1981−1996.
(25) NVIDIA CUDA Parallel Computing Platform. http://www.nvidia.com/cuda/.
(26) Papoulis, A. *McGraw-Hill Series in System Science*; Malabar: Krieger, 1968; p 1.
(27) Hamilton, W. International Tables for X-ray Crystallography; Vol. IV, Table 2.2B. Kynoch Press: Birmingham *(Present distributor Kluwer Academic Publishers, Dordrecht, Table 2.2B)* 1974; pp 273−284.
(28) Marsh, R.; Slagle, K. *Acta Crystallogr., Sect. A: Found. Crystallogr.* **1983**, *39*, 173−173.
(29) Pavlov, M. Y.; Fedorov, B. *Biopolymers* **1983**, *22*, 1507−1522.
(30) Fraser, R.; MacRae, T.; Suzuki, E. *J. Appl. Crystallogr.* **1978**, *11*, 693−694.
(31) Fedorov, B.; Ptitsyn, O.; Voronin, L. *FEBS Lett.* **1972**, *28*, 188−190.
(32) Koutsioubas, A.; Perez, J. *J. Appl. Crystallogr.* **2013**, *46*, 1884−1888.
(33) Schneidman-Duhovny, D.; Hammel, M.; Tainer, J. A.; Sali, A. *Nucleic Acids Res.* **2016**, DOI: 10.1093/nar/gkw389.
(34) Knight, C. J.; Hub, J. S. *Nucleic Acids Res.* **2015**, *43*, gkv309.
(35) Slater, J. C. *J. Chem. Phys.* **1964**, *41*, 3199−3204.
(36) Kahaner, D.; Moler, C.; Nash, S. *Numerical Methods and Software*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1989.
(37) Peter Lepage, G. *J. Comput. Phys.* **1978**, *27*, 192−203.
(38) Shannon, C. E. *Proc. IRE* **1949**, *37*, 10−21.
(39) Beaulieu, N. C. *Proc. IEEE* **2002**, *90*, 276−279.
(40) Farrell, K. W.; Wilson, L. *Biochemistry* **1984**, *23*, 3741−3748.
(41) Ringel, I.; Horwitz, S. B. *J. Pharmacol. Exp. Ther.* **1987**, *242*, 692−698.
(42) Castoldi, M.; Popov, A. V. *Protein Expression Purif.* **2003**, *32*, 83−88.
(43) Sachse, C.; Chen, J. Z.; Coureux, P.-D.; Stroupe, M. E.; Fändrich, M.; Grigorieff, N. *J. Mol. Biol.* **2007**, *371*, 812−835.
(44) Alushin, G. M.; Lander, G. C.; Kellogg, E. H.; Zhang, R.; Baker, D.; Nogales, E. *Cell* **2014**, *157*, 1117−1129.
(45) Díaz, J. F.; Valpuesta, J. M.; Chacón, P.; Diakun, G.; Andreu, J. M. *J. Biol. Chem.* **1998**, *273*, 33803−33810.
(46) Chung, P. J.; Choi, M. C.; Miller, H. P.; Feinstein, H. E.; Raviv, U.; Li, Y.; Wilson, L.; Feinstein, S. C.; Safinya, C. R. *Proc. Natl. Acad. Sci. U. S. A.* **2015**, *112*, E6416−E6425.
(47) DiMaio, F.; Leaver-Fay, A.; Bradley, P.; Baker, D.; André, I. *PLoS One* **2011**, *6*, e20450.

DOI: 10.1021/acs.jcim.6b00159
J. Chem. Inf. Model. 2016, 56, 1518−1527