

Supporting Online Material

$D+$: Software for High-Resolution Hierarchical Modeling of Solution X-Ray Scattering from Complex Structures

AVI GINSBURG,^{a,b} TAL BEN-NUN,^{a,b,c} ROI ASOR,^{a,b} ASAF SHEMESH,^{a,b}

LEA FINK,^{a,b} ROEE TEKOAHA,^{a,b} YEHONATAN LEVARTOVSKY,^{a,b}

DANIEL KHAYKELSON,^{a,b} RAVIV DHARAN,^{a,b} AMOS FELLIG^{a,b} AND URI RAVIV^{a,b*}

^a*Institute of Chemistry, The Hebrew University of Jerusalem, Edmond J. Safra Campus, Givat Ram, 9190401*, ^b*Center for Nanoscience and Nanotechnology, The Hebrew University of Jerusalem, Edmond J. Safra Campus, Givat Ram, 9190401, Jerusalem, Israel*, and ^c*School of Computer Science and Engineering, The Hebrew University of Jerusalem, Edmond J. Safra Campus, Givat Ram, 9190401 Jerusalem, Israel. E-mail: uri.raviv@mail.huji.ac.il*

1. Uniformly spaced 3D Cartesian grid

The naïve approach to span the relevant reciprocal-space would be to use a cube with a side of $2q_{\max}$ from $-q_{\max}$ to q_{\max} . However, because only values within a radius of q_{\max} are needed, the corners need not be allocated and calculated. The ratio between the volumes of the sphere and its enclosing cube is:

$$\frac{\frac{4\pi}{3}q_{\max}^3}{(2q_{\max})^3} = \frac{\pi}{6} \approx \frac{1}{2}$$

Thus, only the points that satisfy

$$|\vec{q}| \leq q_{\max} + 3\Delta q$$

where Δq is the spacing of the grid, should be used. The three extra layers ensure that all points along the shell of radius q_{\max} have calculated points around them for grid interpolation.

2. Finding the rotation angles of an object

To save computations, we bin copies of an object together when their orientation is identical. To find the final rotation angles of an object that has been rotated several times, relative to its initial lab axes directions: $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$, we apply the same rotation on the lab axes unit vector. In other words, we multiply the rotation matrices and obtain the final rotation matrix:

$$\mathbf{XYZ} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \quad (\text{S1})$$

To find the α , β , and γ rotation angles (about the x , y , and z -axes, respectively), in the range between $-\pi$ and π , we compare the \mathbf{XYZ} matrix with the rotation matrix:

$$\mathbf{A}(\alpha, \beta, \gamma) = \begin{bmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta \\ \cos \alpha \sin \gamma + \cos \gamma \sin \alpha \sin \beta & \cos \alpha \cos \gamma - \sin \alpha \sin \beta \sin \gamma & -\cos \beta \sin \alpha \\ \sin \alpha \sin \gamma - \cos \alpha \cos \gamma \sin \beta & \cos \gamma \sin \alpha + \cos \alpha \sin \beta \sin \gamma & \cos \alpha \cos \beta \end{bmatrix}$$

and find the three angles.

To calculate the angles we use the function $\arctan(y, x)$ (the function `ATAN2` in C++), which gets the values of the \cos (for y) and the \sin (for x) of the relevant angle and returns the angle (in radians) in its correct quadrant, based on the sign of the two arguments. If both the \cos and the \sin are positive the angle is between 0 and $\pi/2$. If the \sin is positive and the \cos is negative the angle is between $\pi/2$ and π . If both are negative the angle is between $-\pi$ and $-\pi/2$, and if the \sin is negative and the \cos is positive then the angle is between $-\pi/2$ and 0 .

We then proceed as follows (Heckbert, 1994).

$$\alpha_0 = \arctan(z_2, z_3)$$

We define $c_2 \equiv \sqrt{x_1^2 + y_1^2}$.

If $\alpha_0 > 0$:

$$\alpha_0 = \alpha_0 - \pi$$

$$\beta_0 = \arctan(-z_1, -c_2)$$

else

$$\beta_0 = \arctan(-z_1, c_2)$$

We define:

$$s_1 \equiv \sin(\alpha_0)$$

$$ca_1 \equiv \cos(\alpha_0)$$

and then:

$$\gamma_0 = \arctan(s_1 \cdot x_3 - ca_1 \cdot x_2, ca_1 \cdot y_2 - s_1 \cdot y_3)$$

Finally we get that:

$$\alpha = -\alpha_0, \beta = -\beta_0, \gamma = -\gamma_0.$$

3. PDBUnits: Accessory tool for identifying the rotation and translation of repeating subunits in a PDB file

Complex and large structures often contain repeating subunits that differ from one another in orientation and location. When structures can be represented by hierarchical data tree structures, *D+* is using the Reciprocal Grid (RG) algorithm (Ginsburg *et al.*, 2016). In the data tree structures subunits are leaves and the manner by which the subunits are organized is represented by assembly symmetry nodes (Fig. 1). In the RG method the scattering amplitude of the assembly is calculated by computing the

amplitudes of the basic subunits (leaves) on $3D$ reciprocal-space grids, moving up in the hierarchy, calculating the grids of larger structures containing repeats of the subunits. By repeating this process for all the leaves and nodes of the tree the scattering amplitude of the final complex structure is obtained. In this way, there is no need to recalculate the scattering of each instance of a repeating subunit. Therefore, to reduce the computation time when dealing with large supramolecular structures, it is best to identify repeating subunits. *D+* allows users to import `Manual Symmetries`, which describe the manner by which repeating subunits are rotated and shifted in space. To efficiently compute the scattering amplitude from a Protein Data Bank (PDB) file of a complex structure made of many repeating subunits using the RG algorithm, *D+* requires a list of all the orientations and translations (or symmetries) of all the instances of each repeating subunit. The PDBUnits tool, which will be described in the following, can automatically export the symmetry of each type of subunit. The input is the complete structure (represented by a PDB file), the structure of a subunit (represented by a smaller PDB file) at a specific orientation (that we shall call the “original” orientation), and the tolerance, given by the maximum root-mean-squared-displacement (RMSD) value within which repeating subunits can be considered similar. The program reads the complete structure and finds all the instances of the subunit. For each instance, the rotation and translation with respect to the original subunit is computed. To ensure consistency, the rotation and translations are applied to the original subunit and the RMSD between the original subunit and the examined instance of the subunit is computed. A comparison is made between the examined instance and a duplicated subunit, obtained by applying the translation and rotation that best fitted the examined subunit, to its original orientation. This process can be repeated for every additional similar subunit that the complete structure contains. The results are exported to *D+* input format for immediate use.

3.1. Implementation

The program gets the PDB file of the complete structure and the PDB of the inspected subunit. It reads the files and converts them into arrays of atomic symbols and their real-space location coordinates. The array of the complete structure is called \mathbf{A} and the array of the subunit is called \mathbf{S} . \mathbf{A}_i corresponds to the i th atom in array \mathbf{A} . The first condition for identifying a group of atoms in \mathbf{A} as an instance of \mathbf{S} is having an identical atomic sequence to \mathbf{S} . The program runs through array \mathbf{A} and identifies all the \mathbf{S} atom sequences. Each sequence of atoms in \mathbf{A} with atomic sequence \mathbf{S} is called \mathbf{P} (as this set is a potential instance of \mathbf{S}). Being identical to the atomic sequence of \mathbf{S} is a necessary but not a sufficient condition for \mathbf{P} to be identified as an instance of the subunit \mathbf{S} .

To further explore the character of \mathbf{P} we need to find its translation and rotation in relation to \mathbf{S} . To find the rotation of \mathbf{P} with respect to \mathbf{S} , we need to translate both to the origin:

$$\vec{P}_i = B\vec{P}_i - \vec{P}_{\text{cm}}$$

$$\vec{S}_i = B\vec{S}_i - \vec{S}_{\text{cm}}$$

This operation is applied to each atom in \mathbf{P} and \mathbf{S} where \vec{P}_i and \vec{S}_i are the i th atom vectors after translation and, $B\vec{P}_i$ and $B\vec{S}_i$ are the i th atom vectors before translation ($i \in \{1, \dots, N_p\}$, where N_p is the number of atoms in each array) and \vec{P}_{cm} and \vec{S}_{cm} are the centers of mass of each structure respectively.

The tool then creates two reference matrices from the first four atoms of \mathbf{P} and \mathbf{S} :

$$\mathbf{M}_P = \begin{bmatrix} \vec{P}_1 - \vec{P}_0 \\ \vec{P}_2 - \vec{P}_0 \\ \vec{P}_3 - \vec{P}_0 \end{bmatrix},$$

$$\mathbf{M}_S = \begin{bmatrix} \vec{S}_1 - \vec{S}_0 \\ \vec{S}_2 - \vec{S}_0 \\ \vec{S}_3 - \vec{S}_0 \end{bmatrix}.$$

\mathbf{M}_P is the reference matrix for \mathbf{P} and \mathbf{M}_S is the reference matrix for \mathbf{S} . The aim is to find the rotation matrix, \mathbf{R} , that can rotate \mathbf{M}_S to \mathbf{M}_P :

$$\mathbf{M}_P = \mathbf{R} \cdot \mathbf{M}_S.$$

By multiplying both sides by \mathbf{M}_S^{-1} we get:

$$\mathbf{R} = \mathbf{M}_P \cdot \mathbf{M}_S^{-1}.$$

The program computes \mathbf{R} . If \mathbf{P} is an instance of the \mathbf{S} , then each atom in \mathbf{P} should be obtained by rotating the corresponding atom in \mathbf{S} by \mathbf{R} . If any atom in \mathbf{P} has a different rotation with respect to \mathbf{S} , then \mathbf{P} is not an instance of \mathbf{S} . In that case, \mathbf{P} and \mathbf{S} have identical atomic sequence but different conformations. The tool rotates a copy of \mathbf{S} by \mathbf{R} :

$$\mathbf{RS} = \mathbf{R} \cdot \mathbf{S}.$$

If \mathbf{RS} is overlapping with \mathbf{P} , then \mathbf{R} describes the rotation of \mathbf{S} to \mathbf{P} . Because PDBs are based on experimental data, and because of numerical artifacts, \mathbf{RS} does not always accurately overlap with \mathbf{P} , even if \mathbf{P} is indeed an instance of \mathbf{S} . To take this into account, the program calculates the $RMSD_{\text{rotation}}$ between \mathbf{RS} and \mathbf{P} :

$$RMSD_{\text{rotation}} = \frac{1}{N_p} \sum_{i=1}^{N_p} \sqrt{(\vec{RS}_i - \vec{P}_i)^2}$$

If the $RMSD_{\text{rotation}}$ exceeds a given tolerance value, ϵ , the tool does not count \mathbf{P} as an instance of \mathbf{S} and continues to search for other instances. If $RMSD_{\text{rotation}} \leq \epsilon$, then the tool continues to check the translation of \mathbf{P} with respect to \mathbf{S} .

In the complete structure (**A**), an instance of the subunit (**S**) can be translated. In order to find that translation-vector, **S** is rotated by the matrix **R** that was found above, $\mathbf{RS} = \mathbf{R} \cdot \mathbf{S}$. By rotating **S** to the orientation of **P**, the tool eliminates the effect of the rotation on the coordinates and examines the translation alone:

$$\vec{T} = \vec{P}_{\text{cm}} - \vec{R}\vec{S}_{\text{cm}}$$

\vec{T} is the translation vector, \vec{P}_{cm} is the center of mass of **P** and $\vec{R}\vec{S}_{\text{cm}}$ is the center of mass of **RS**. If **P** is an instance of **S**, then \vec{T} describes the translation relations of all of the atoms of **P** with respect to **RS**. The tool then translates all the atoms of **RS** by the vector \vec{T} :

$$\mathbf{TS} = \mathbf{RS} + \vec{T}$$

If **P** is an instance of **S**, **TS** should overlap with **P** to within an $RMSD_{\text{translation}}$ smaller than ϵ , where:

$$RMSD_{\text{translation}} = \frac{1}{N_p} \sum_{i=1}^{N_p} \sqrt{(\vec{T}\vec{S}_i - \vec{P}_i)^2}$$

If $RMSD_{\text{translation}} \leq \epsilon$, **P** is considered an instance of **S**, otherwise the tool does not count **P** as an instance of **S** and continues to search for other instances. The process that was described above is repeated for every group of atoms that their atomic sequence is identical to **S**. In the end, the tool recognizes all of the instances of subunit **S** in **A**.

3.1.1. Format and units Docking list (DOL) text file is the input format for manual symmetries in *D+*, and is therefore, the output format of the tool. The tool writes every instance of subunit, **S**, to a DOL file that can be later used by *D+*. To match the format of the DOL file, the translation vector which is in PDB units (Å) is divided by 10, to match the units of *D+* (nanometers). The tool extracts the angles from the rotation matrix by Tait-Bryan rotation angle convention used in *D+* (in which the

rotation matrix is ordered as **XYZ**). The Tait-Bryan rotation angles are in radians. *D+* uses angles in degrees in the DOL file. The Tait-Bryan rotation angles are therefore converted into degrees before writing them into the DOL file.

3.2. Test examples

A small CCl_4 subunit in tetrahedron symmetry was created manually as a test case. The subunit was placed in a PDB twice, in different orientations and translations. It was placed along with random atoms and some atomic sequences that are similar but not identical to the subunit. Two identical atomic sequences were also placed (C and 4 Cl) but as separate atoms and not in a structure (to verify that the tool successfully discards them). The tool loaded the PDB of CCl_4 and of the complete structure. The tool successfully recognized the two instances of CCl_4 and discarded any other similar but wrong cases. A DOL file was exported. *D+* loaded the DOL file and the PDB of the subunit, and a scattering curve was computed. A PDB containing the two CCl_4 orientations only was created by removing the random atoms and misleading structures and its scattering curve was computed by *D+*, using a **Grid Size** of 100 (Fig. S1A).

A second example, is the PDB 3J6F (Alushin *et al.*, 2014), containing 9 tubulin dimers from the microtubule (MT) structure. A PDB representation of a single tubulin dimer subunit was extracted from 3J6F. The two PDB files were loaded to the tool. The tool successfully recognized all the 9 tubulin dimers and exported the translations and rotations into a DOL file. The DOL file was loaded to *D+* along with the tubulin dimer subunit PDB and the scattering signal was computed. The scattering curve of 3J6F PDB was also computed by *D+*, using a **Grid Size** of 200 (Fig. S1B).

The last example is of tubulin ring comprising 13 tubulin dimers. The PDB of the full ring structure and the PDB of a tubulin dimer were loaded to our tool. The tool successfully identified the 13 repeating dimer subunits and exported a correct DOL

file. *D+* computed the scattering curves from the PDB of the full ring or from the DOL file and the PDB of the dimer, using a **Grid Size** of 300 (Fig. S1C).

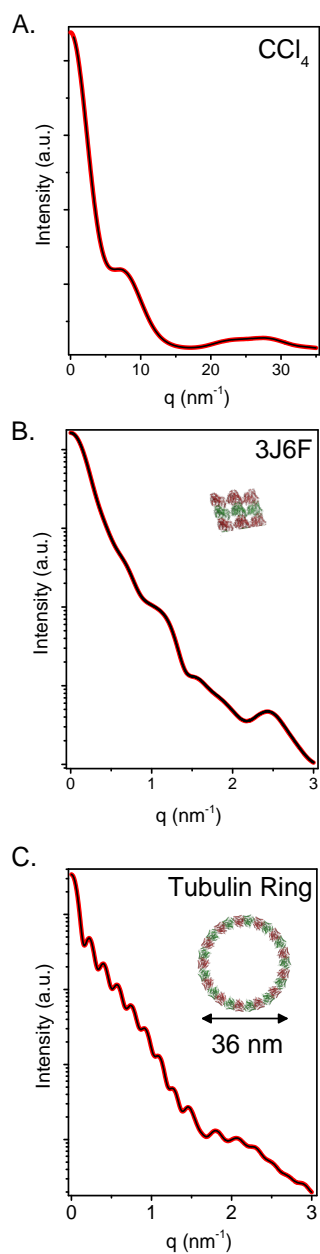


Fig. S1. Modeled scattering intensity as a function of the magnitude of the scattering vector, q . The scattering curves were computed by $D+$ using the subunit PDB file and the DOL file obtained from our PDBUnits tool, using ϵ of 0.01 \AA (red curves), or the PDB file of entire structure (black curves). **A.** CCl_4 **B.** Nine tubulin dimers from the MT structure. **C.** Tubulin ring comprising 13 dimers. In all cases, $D+$ used a Convergence tolerance of 0.001.

Fig. S1 shows that scattering curves that were computed with the DOL that was created by the tool, completely overlapped with the curves that were computed from the PDB of the entire assembly. In addition, the PDB files that were exported from $D+$ using the DOL files and the PDB of the subunit were identical to the original full PDB. The computation time was about ten times shorter when DOL files were used owing to the advantages of the RG algorithm (Ginsburg *et al.*, 2016). Therefore, whenever there are repeating subunits, especially in large assemblies, it is recommended to analyze the structure with the tool and only then by $D+$.

4. Finding the principal axes of an atomic structure

Given a $3 \times n$ matrix containing the coordinates of n atoms in a PDB file (whose origin is at the center of mass):

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix},$$

the principal axes can be found (as in principle component analysis (Jolliffe, 2002)), in the following manner. Matrix \mathbf{M} is defined by:

$$\mathbf{M} = \mathbf{A}^T \mathbf{A}.$$

\mathbf{E} is the eigenvector (\vec{E}_i) matrix of \mathbf{M} , sorted in descending order of the eigenvalues (e_i). If the object is more prolate, i.e. $(e_1 - e_2) < (e_2 - e_3)$ then we align e_3 with the z -axis. The rotation axis is given by $\hat{\mathbf{r}}\mathbf{a} = \frac{\hat{z} \times \vec{E}_3}{\|\vec{E}_3\|}$ and the angle is:

$$\theta = \arccos \left(-\frac{\hat{z} \cdot \vec{E}_3}{\|\vec{E}_3\|} \right).$$

The rotation matrix is:

$$\mathbf{R} = \begin{bmatrix} ra_x^2(1 - \cos \theta) + \cos \theta & ra_x ra_y(1 - \cos \theta) - ra_z \sin \theta & ra_x ra_z(1 - \cos \theta) + ra_y \sin \theta \\ ra_x ra_y(1 - \cos \theta) + ra_z \sin \theta & ra_y^2(1 - \cos \theta) + \cos \theta & ra_y ra_z(1 - \cos \theta) - ra_x \sin \theta \\ ra_x ra_z(1 - \cos \theta) - ra_y \sin \theta & ra_y ra_z(1 - \cos \theta) + ra_x \sin \theta & ra_z^2(1 - \cos \theta) + \cos \theta \end{bmatrix}$$

where ra_x, ra_y , and ra_z are the elements of $\hat{\mathbf{r}}\mathbf{a}$. If the object is more oblate i.e. $(e_1 - e_2) > (e_2 - e_3)$, \vec{E}_1 is aligned to the z -axis using $\hat{z} \times \vec{E}_1$. Aligning the second axis is done by setting $\mathbf{A}' = \mathbf{R} \cdot \mathbf{A}$ and repeating the process using $\hat{y} \times \vec{E}_2$.

5. Supernatant subtraction from microtubule sample

Dynamic microtubule (MT) was assembled in the presence of 4 mM GTP and no stabilizing agent was added. The MT sample contained MTs as well as other smaller tubulin aggregates. By centrifuging the sample (20, 800 g, at 36 °C for 30 min), the MT were removed to a pellet and the remaining supernatant was measured and used as background for the MT sample. The resulting subtracted curve has identifiable features throughout the entire scattering range (Fig. S2).

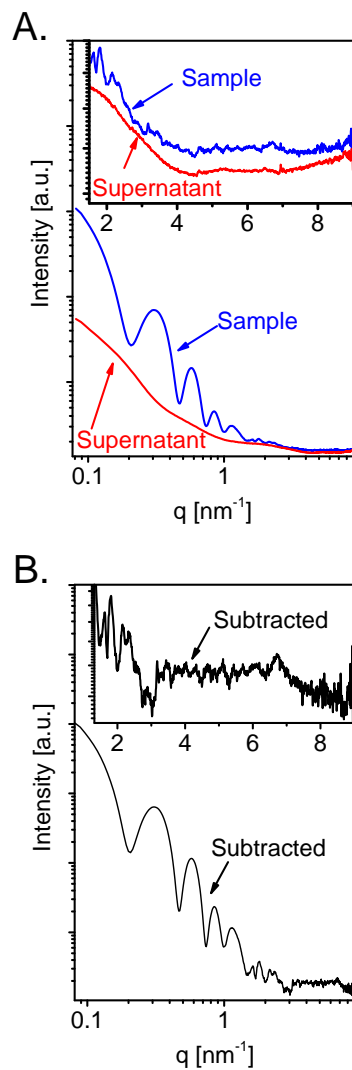


Fig. S2. Azimuthally integrated scattered intensity curves as a function of the magnitude of the scattering vector, q , from the dynamic microtubule (MT) sample in Fig. 6, its supernatant, and the subtracted curve. **A.** The blue curve is the scattering from the MT sample, containing MT, tubulin dimer, and small tubulin assemblies. The red curve is the scattering from the supernatant after centrifuging the sample. **B.** The black curve is the resulting subtracted signal (sample minus supernatant). The insets show the high q -range data on an expanded scale.

6. Comparing D+ with CRY SOL

Fig. S3A compares the computed scattering curves from the atomic model of lysozyme in water, based on PDB ID 1LYZ, where hydrogen atoms were implicitly taken into account. The curves were computed by *D+*, *DebyeCalculator*, and CRY SOL. In Fig. S3B, a similar comparison was done after hydrogen atoms were explicitly added to the PDB file (using PyMol software (Schrödinger, 2015)). Fig. S3C, compares the computation results of the two programs when a solvation layer was added. We attribute the differences in the results to the slightly different algorithms used by the two programs. In Fig. S3A and B, the computations were also compared with *DebyeCalculator*, which applies Debye formula, and is in good agreement with *D+*.

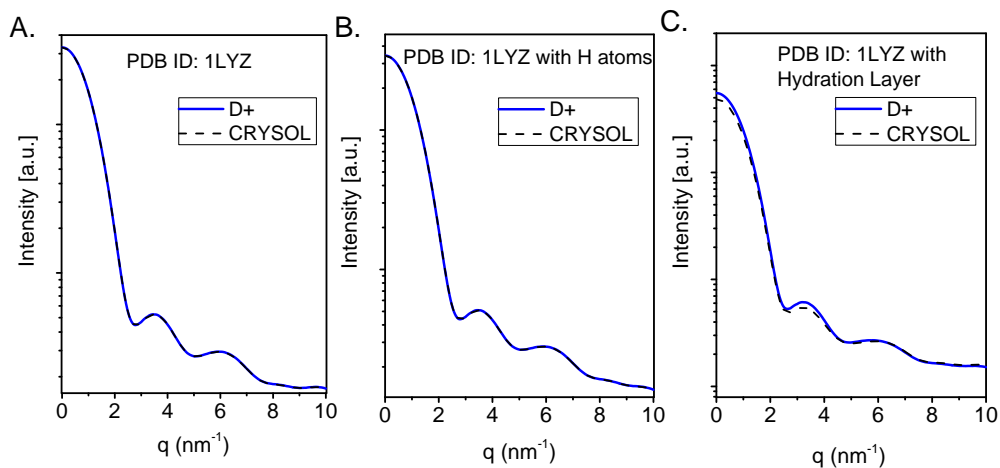


Fig. S3. Computed scattered intensity curves as a function of q , from the atomic models of lysozyme (PDB ID 1LYZ). The dashed black curve was computed by CRY SOL (ATSAS 2.8.2) (Svergun *et al.*, 1995), where the maximum order of harmonics was 50 and the order of Fibonacci grid was 18. The solid blue curves were computed by *D+*. In both programs, the solvent density of water was set to $334 \text{ e} \cdot \text{nm}^{-3}$. **A.** The PDB structure in water using implicit hydrogen atoms. **B.** Adding hydrogen atoms to the PDB using PyMOL (Schrödinger, 2015) and computing the scattering in water with explicit hydrogen atoms. **C.** The solvation layer was taken into account, using the default parameters of both programs: solvent Probe Radius of 0.14 nm, solvation layer thickness of 0.3 nm with an electron density contrast of $30 \text{ e} \cdot \text{nm}^{-3}$. Intensity curves computed by *DebyeCalculator* gave similar results to those of *D+* in **A** and **B**. Solvation shells cannot be added in *DebyeCalculator*.

7. Performance Analysis

Figure S4 shows the number of spherical harmonics expansion terms, lm , in CRY SOL (ATSAS 2.8.2) (Svergun *et al.*, 1995) as a function of the number of tubulin dimers, N , in the models. The reported lm is the minimum value need to obtain residuals (Eq. S2) of 0.05 or less at each q point between CRY SOL (ATSAS 2.8.2) and D+ or *DebyeCalculator*, which uses Debye formula (Debye, 1915). For $N \geq 43$ the maximum lm value of CRY SOL (99), was insufficient to meet accuracy criterion. Intensity curves were computed, in both D+ and CRY SOL, between $q_{\min} = 0$ and $q_{\max} = 5 \text{ nm}^{-1}$.

Fig. S5 repeats the analysis of Fig. 7 using either Grids or the Hybrid methods. When the number of dimers is below 140 (in most cases) Grids are faster than Hybrid. When the structures contain more dimers, the Hybrid method is only possible solution. Fig. S6 repeats the performance analysis of Fig. 7 and computes the contribution of the solvation layer using a much smaller `Solvent Voxel Size` of 0.05 nm (instead of 0.2 nm. The latter size is smaller than used by other programs (Grudin *et al.*, 2017; Svergun *et al.*, 1995)). Computations with `Solvent Voxel Size` of 0.05 nm are slower but can be more accurate in some cases. Fig. S7 directly compares the effect of `Solvent Voxel Size` within D+, showing that at low number of dimers the effect is more pronounced. Fig. S8 show larger `Grid Size` extends the computation times, when using MC or Vegas MC orientation average integration methods. In the case of Gauss Kronrod the convergence was faster when the Grids were denser. Fig. S8 uses grids in all the computations. Fig. S9 show a similar comparison when using the Hybrid method. In the latter case, the effect of `Grid Size` is similar but much smaller.

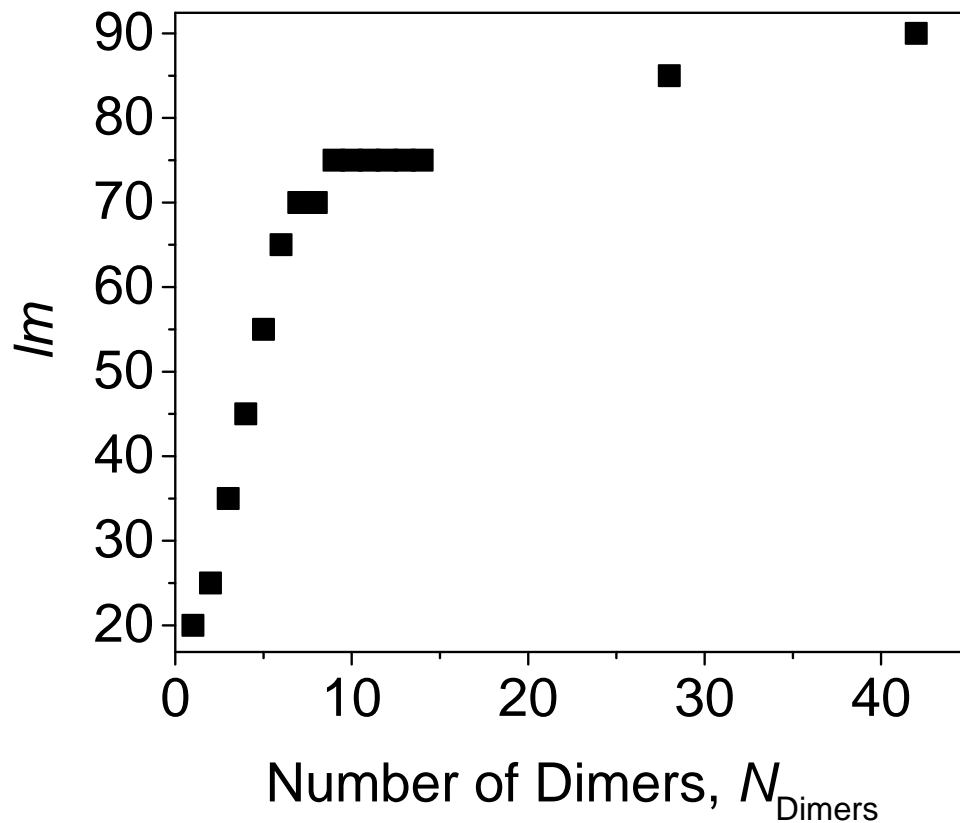


Fig. S4. The minimum number of spherical harmonics expansion terms, l_m , in CRY SOL (ATSAS 2.8.2) (Svergun *et al.*, 1995), which were required to attain accurate results in Fig. 7, as a function of the number of tubulin dimers, N_{Dimers} . The level of accuracy was set to have a residual value (Eq. S2) of 0.05 or less, compared with either *DebyeCalculator* or D+. Intensity curves were computed, in both D+ and CRY SOL, between $q_{\text{min}} = 0$ and $q_{\text{max}} = 5 \text{ nm}^{-1}$.

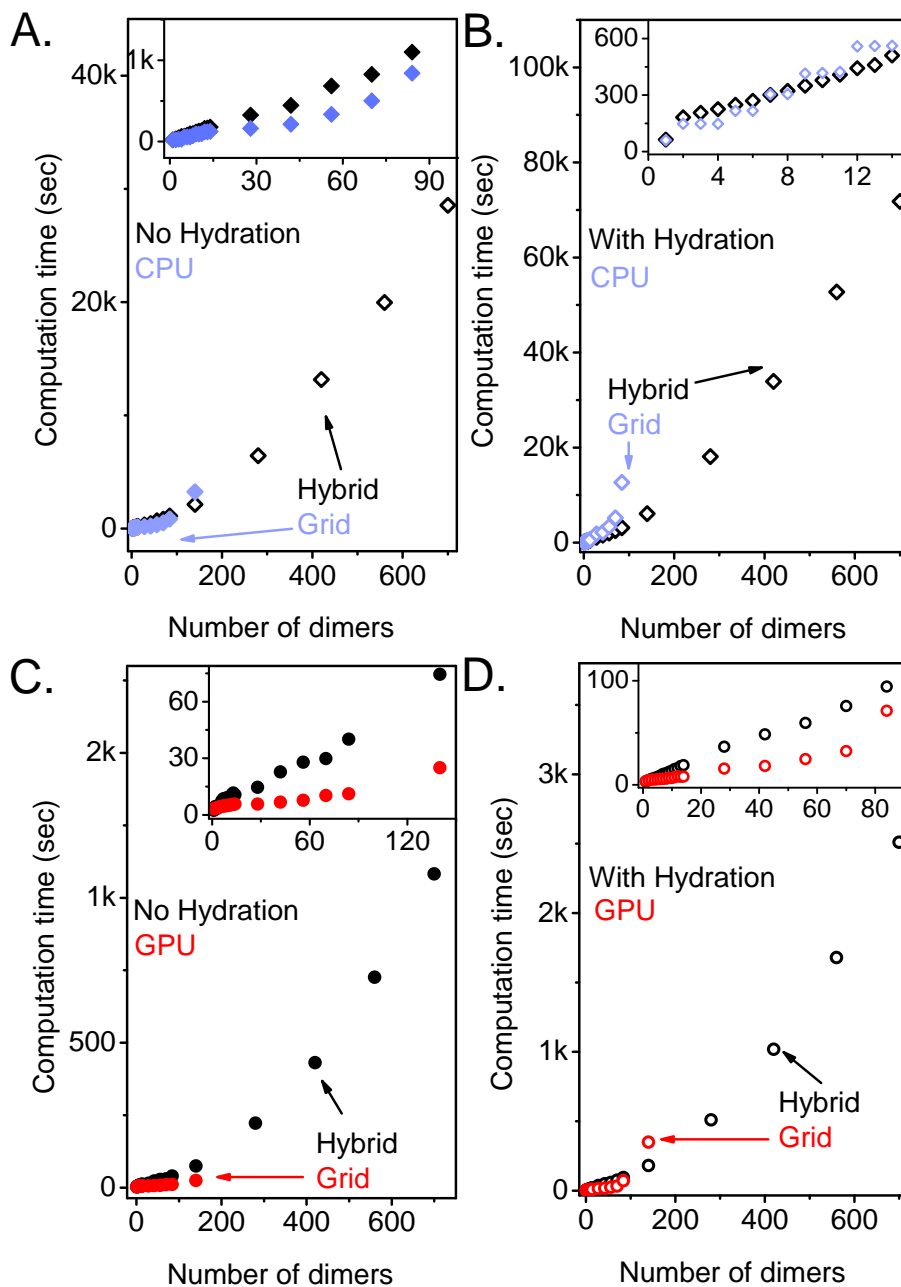


Fig. S5. Comparing Grids with Hybrid. Repeating the analysis of Fig. 7 using Grids up to 140 dimers. Grid Size as a function of the number of dimers was as in Fig. 7D Red symbols indicate GPU computations. Light blue symbols indicate CPU computations. Black symbols corresponds to computations with the Hybrid method, using a Grid size of 40 when no solvation layers were computed (A and C), or 60 when solvation layers were taken into account (B and D).

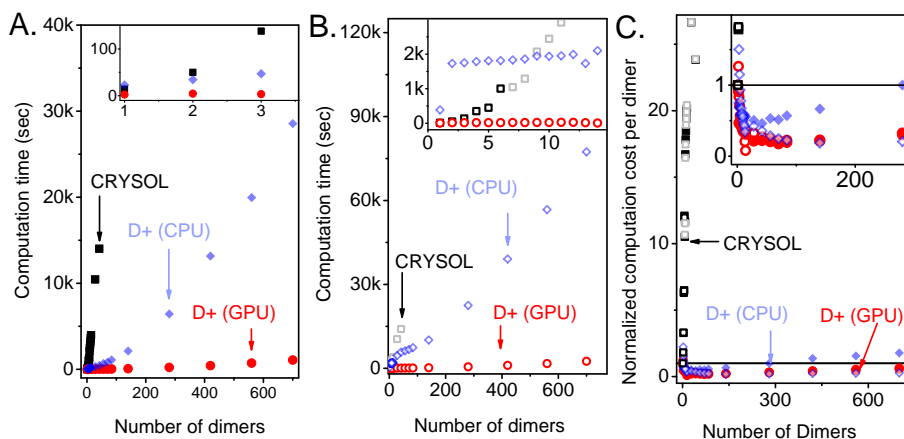


Fig. S6. Performance analysis as in Fig. 7, using a much smaller Solvent Voxel Size of 0.05 nm (instead of 0.2 nm). For structures with more than one dimer, the Hybrid method of D+ was used with a Grid size of 40 when no solvation layer was computed (in A), and 60 when solvation layer was taken into account (in B).

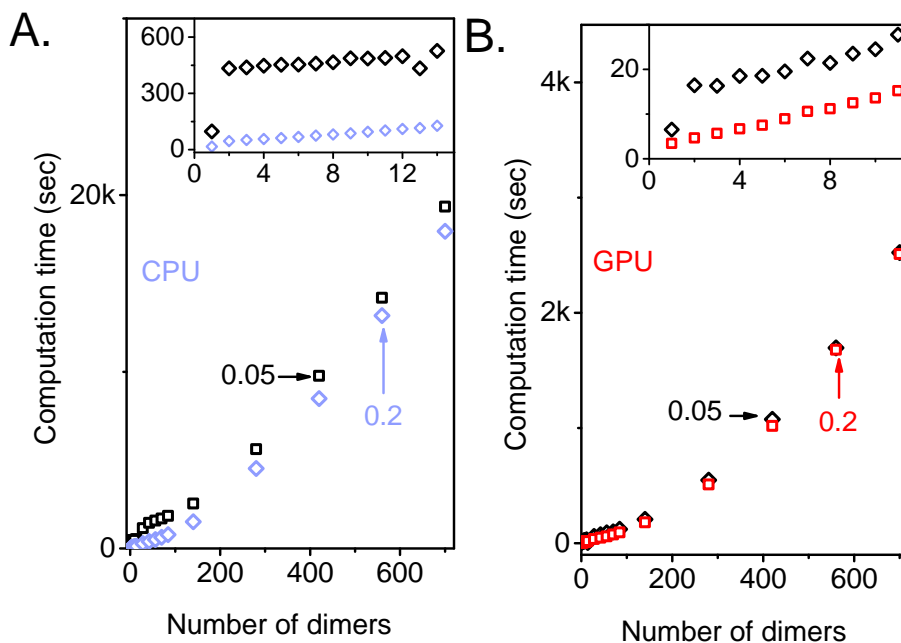


Fig. S7. Effect of Solvent Voxel Size in D+. Direct comparison of the computations times in Fig. 7B and Fig. S6B. **A.** CPU computation times using Solvent Voxel Size of 0.2 or 0.05. **B.** GPU computation times using Solvent Voxel Size of 0.2 or 0.05. For structures containing more than a dimer, the Hybrid method was used with a Grid size of 60.

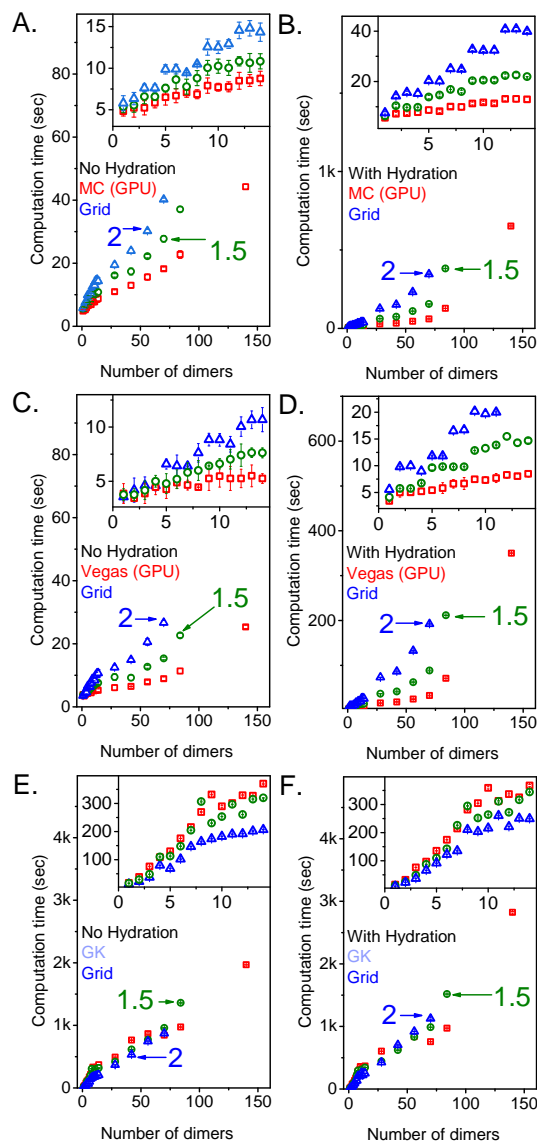


Fig. S8. Effect of Grid Size when using grids only. Repeating the computations in Fig. 7 using the optimal Grid Size shown in Fig. 7D (red symbols), or 1.5 times the optimal Grid Size (olive symbols) or twice the optimal Grid Size (blue symbols). **A.** and **B.** Monte Carlo integration using GPU. **C.** and **D.** Vegas Monte Carlo integration using GPU. **E.** and **F.** Gauss Kronrod integration, using GPU for the grid computations and CPU for the orientation average. Solvent Voxel Size was 0.2 in **B,** **D,** and **F.**

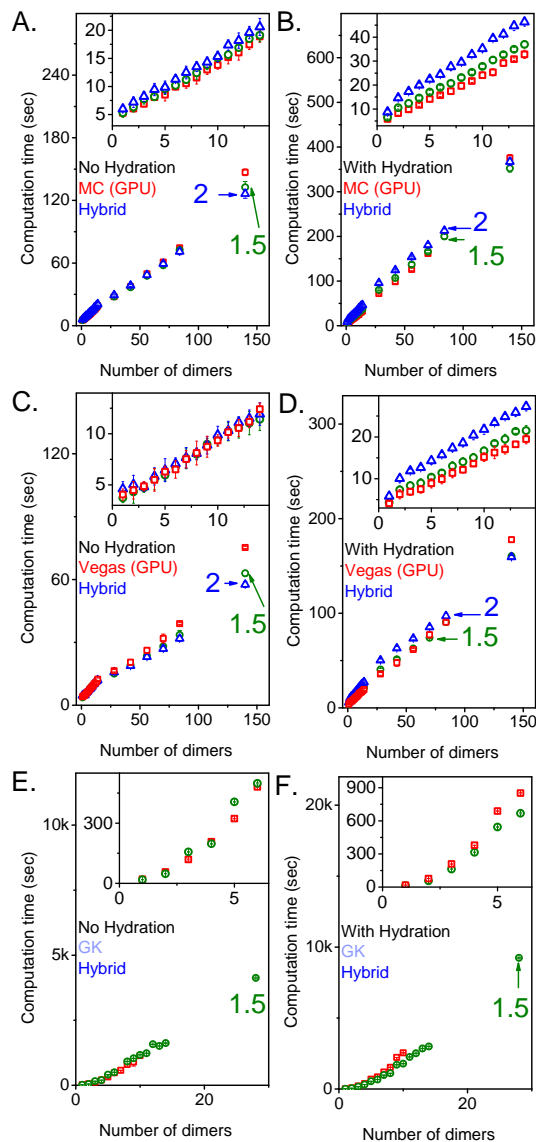


Fig. S9. Effect of **Grid Size** when using the Hybrid Method. Repeating the computations in Fig. S8 using the Hybrid method. The optimal **Grid Size** was 40 in **A**, **C**, and **E** and 60 in **B**, **D**, and **F** (red symbols). Olive symbols indicate 1.5 times the optimal **Grid Size**. Blue symbols indicate twice the optimal **Grid Size**. **A.** and **B.** Monte Carlo integration using GPU. **C.** and **D.** Vegas Monte Carlo integration using GPU. **E.** and **F.** Gauss Kronrod integration, using GPU for the grid computations and CPU for the orientation average. **Solvent Voxel Size** was 0.2 in **B**, **D**, and **F**.

8. Comparing experimental microtubule solution scattering data to microtubule models with varying protofilament number

Fig. S10 compares the experimental dynamic MT scattering data with different atomic MT models whose protofilament number vary between 12 and 15.

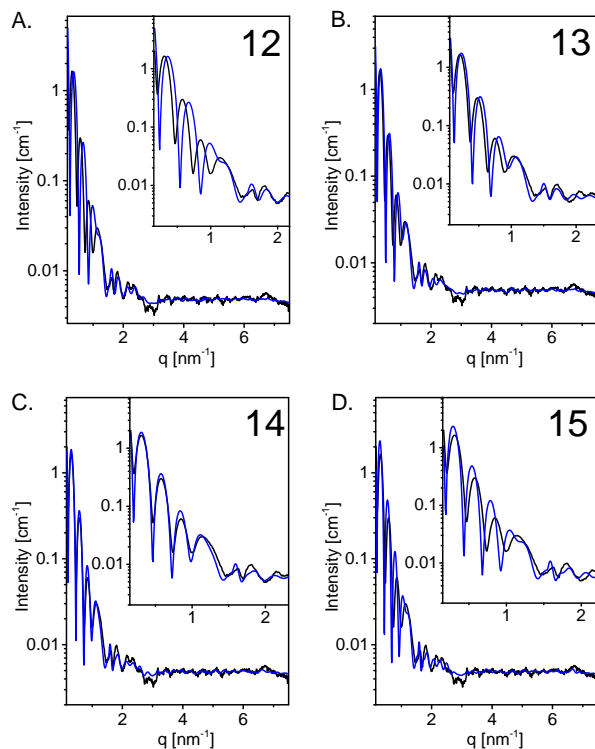


Fig. S10. Solution X-ray scattering data from dynamic microtubule (MT) and MT models with varying number of protofilaments. Azimuthally integrated, background-subtracted, experimental scattering intensity from dynamic MT (black curves), taken from Fig. S2B. The blue solid curves were computed from atomic MT models with 12 (A), 13 (B), 14 (C) or 15 (D) protofilaments, each containing 48 tubulin dimers along the long MT axis. The dimers were arranged in a 3-start left-handed helical lattice, whose radii were 10.2, 11.05, 11.9, and 12.75 nm, corresponding to 12, 13, 14, and 15 protofilaments, respectively. The pitch in all the models was 12.214 nm, in agreement with PDB 3J6F (Alushin *et al.*, 2014). A solvation layer was added to the MT models, using a Probe Radius of 0.14 nm and the solvation overlap correction method, described in §10.2. The layer thickness was 0.28 nm and its electron density was $364 \text{ e} \cdot \text{nm}^{-3}$. The inset shows the low q -range on an expanded scale. Intensity curves were computed in about half a minute on an NVIDIA Titan GPU, using the Hybrid method.

9. Validation tests of D+

D+ has been extensively tested, using the hardware architecture described in §6.1.3, in order to ensure that its computations are correct and consistent with other programs. In §9.1, geometric models were compared with X+ (Ben-Nun *et al.*, 2010), which uses analytical expressions when possible. In §9.2, atomic models were compared with CRY SOL (ATSAS 2.8.2) (Svergun *et al.*, 1995) or *DebyeCalculator*, which uses Debye formula (Debye, 1915). Computations in CRY SOL were done using its maximum accuracy settings, including 99 as the order of spherical harmonics terms and 18 as the order of the Fibonacci grid (when relevant). In §9.3, the geometric models were cross-validated using atomic structures that assumed the corresponding geometric shapes.

In addition, various cross validation tests were designed to check that D+ is consistent within itself by constructing the same structures using different options in D+ and comparing the results. We present several examples that demonstrate how D+ was tested. The full set of tests is larger and provided as an additional supporting folder that is part of the software package. In each test, we show comparable curves and a zoom into a section where small deviations between curves can be observed. At each q value, we also present the normalized residuals:

$$R(q) = \frac{I_{test}(q) - I_{ref}(q)}{I_{test}(q)}. \quad (\text{S2})$$

I_{test} is the tested intensity, computed by D+. I_{ref} is the reference curve, computed by CRY SOL (ATSAS 2.8.2) (Svergun *et al.*, 1995), or X+ (Ben-Nun *et al.*, 2010), or *DebyeCalculator*, or by D+ using an alternative way of computation. The differences between the intensities are normalized to the magnitude of the intensity. Although I_{test} is close to I_{ref} , the analytical results of X+ may lead to values of I_{ref} , which can be very close to zero. We therefore normalized the residuals by I_{test} rather than

I_{ref} . In those cases, we plotted the normalized absolute residuals, $|R|$, on logarithmic scales, which are suitable to present a wide range of residual magnitudes. The wide range was owing to the sharp minima in the scattering curves where the normalized absolute residuals approached unity, whereas the other parts of the curves approached zero. Some of the tests can be found in `./Example Files/` directory, where D+ is installed.

9.1. Geometric models

9.1.1. Cylindrical models Fig. S11 compares between the scattering curve from a model of a single 100 nm long rod with a radius of 1 nm, computed in X+, with a similar model, computed in D+, using a **Grid Size** of 330. The same model was then constructed in D+ from 10 identical 10 nm long rods with the same radius, placed on top of one another, using a space-filling 1D symmetry (see cartoon in Fig. S11). In the second computation, the **Grid Size** of the small rod was only 50 and the Hybrid method was used to compute the Space-Filling symmetry. The results were very similar up to negligible small numerical errors.

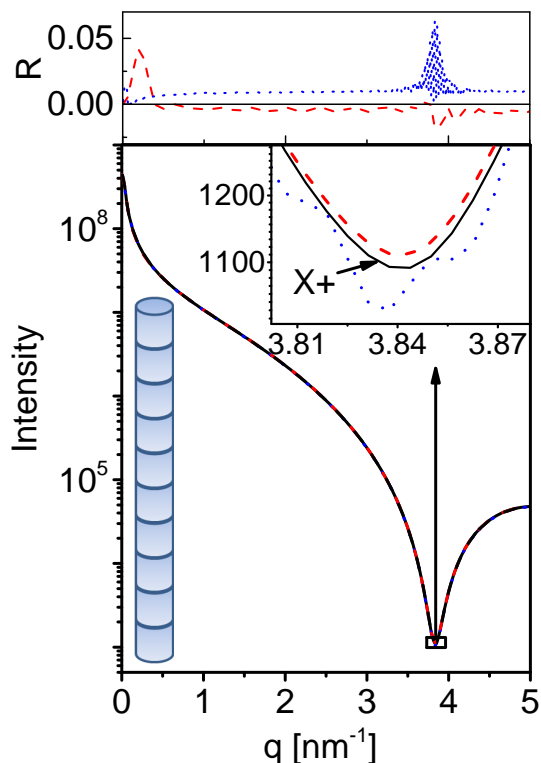


Fig. S11. Modeled X-ray scattering intensity curves from 100 nm long uniform rod with a radius of 1 nm and electron density of 400 e/nm^3 , computed in X+ (solid black curve) or in D+ using GPU, adaptive (VEGAS) Monte Carlo integration and **Grid Size** of 330 (dashed red curve). The same structure was computed in D+ by computing the form-factor of 10 nm long rod and placing it in a Space-Filling symmetry containing 10 identical units on top of one another (see cartoon). The **Grid Size** of the short rod was 50 and the Space-Filling symmetry was computed using the Hybrid method (dotted blue curve). The inset presents, on an expanded scale, a small part of the results where the deviation between the methods is large. The top curve shows the normalized residuals, R , where the reference curve is the intensity computed by X+ (Eq. S2).

Fig. S12 compares between models of uniform concentric cylinders of finite height computed in X+, with those computed in D+. Within D+, the same models were computed in several different ways. All the computations gave very similar results.

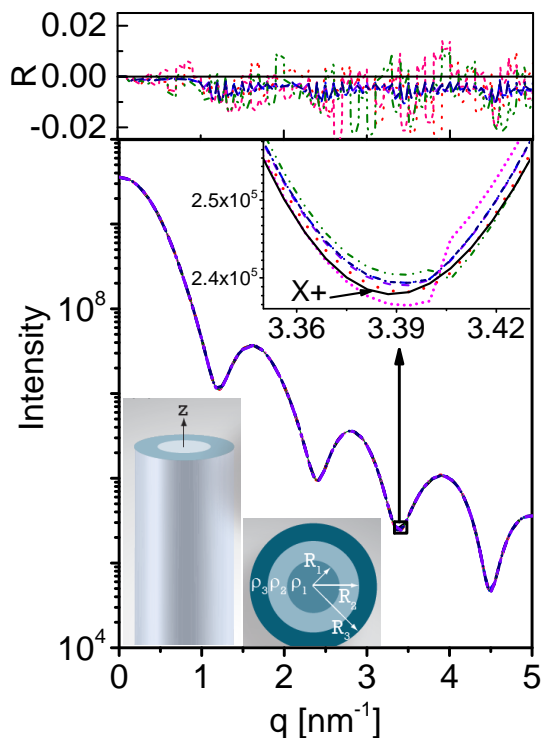


Fig. S12. Solution X-ray scattering intensity from 10 nm long uniform rod surrounded by two concentric hollow cylinders. The radius of the rod is 1 nm and its electron density is 400 e/nm^3 . The second shell is 1 nm thick and its electron density is 500 e/nm^3 . The third shell is 1 nm thick and its electron density is 600 e/nm^3 . The solid black curve was calculated with X+. The other broken curves were computed by D+ using different options as follows. The model was constructed using a single cylindrical geometry with three shells (as in X+) and was computed either on the CPU (olive), using adaptive Gauss-Kronrod integration, or on the GPU (navy), using adaptive (VEGAS) Monte Carlo integration. We then created the same structure using a rod and two separate concentric hollow cylinders with one shell each. The three models (leaves in the data tree structure) were then positioned in a concentric symmetry using the Manual Symmetry option of D+. The models were then calculated on CPU (pink) using the adaptive Gauss Kronrod integration method and on the GPU (blue) using the adaptive (VEGAS) Monte Carlo integration option. The same structure was constructed from two 5 nm long concentric cylinders that were placed on top of one another using Manual Symmetry (CPU-violet and GPU - red). The inset shows a small part of the curves on an expanded scale. The cartoon shows a side-view and a top-view of the computed models. The top curves show the normalized residuals, R , where the reference curve is the intensity computed by X+ (Eq. S2).

9.1.2. *Spherical models* Fig. S13 compares between models of uniform concentric spheres computed in X+ with those computed in D+. Within D+, the same models were computed in several different ways. All the computations gave very similar results.

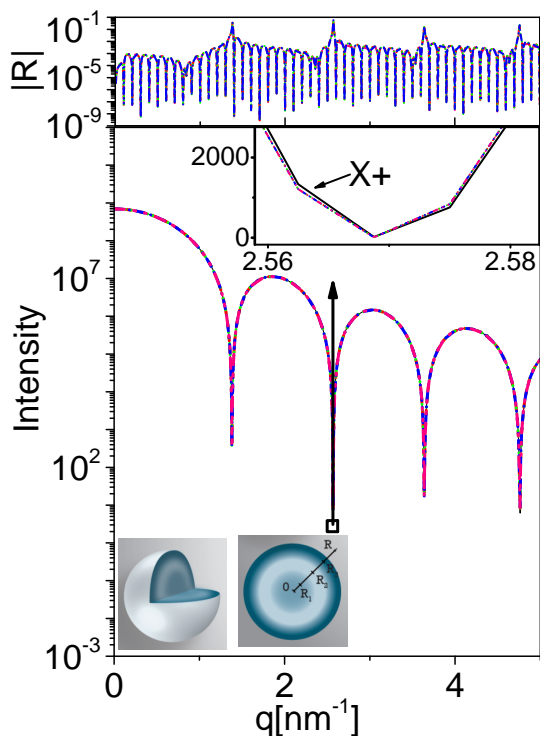


Fig. S13. Modeled X-ray scattering intensity from a uniform sphere surrounded by two concentric uniform hollow spherical shells. The radius of the core sphere is 1 nm and its electron density is $400 \text{ e}/\text{nm}^3$. The first spherical shell is 1 nm thick, its inner radius is 1 nm, and its electron density is $500 \text{ e}/\text{nm}^3$. The second spherical shell is 1 nm thick, its inner radius is 2 nm, and its electron density is $600 \text{ e}/\text{nm}^3$. The solid (black) curve was computed in X+. The other broken curves were computed by D+, using different options as follows. The model was constructed using a single spherical geometry with three layers (as in X+) and was computed either on the CPU (green) and GPU (pink), using Monte Carlo integration. We then created the same structure using a sphere and two separate concentric hollow spherical shells. The three models (leaves in the data tree structure) were then positioned in a concentric symmetry using the Manual Symmetry option of D+. The models were then computed using CPU (blue) or GPU (red), using Monte Carlo integration. The inset shows a small part of the curves on an expanded scale. The cartoons show a side view and a cut through the computed structures. The top curves show the normalized absolute residuals, $|R|$, where the reference curve is the intensity computed by X+ (Eq. S2).

9.1.3. Slab models Fig. S14 compares between a model of an infinite multi-layered slabs, computed by X+, with similar symmetric slab models computed by D+, using finite size structures. A $100 \times 100 \text{ nm}^2$ multi-layered slab was computed by the Direct method. Then, a $1,000 \times 1,000 \text{ nm}^2$ multi-layered slabs was computed by constructing each slab from 10,000 slabs of $10 \times 10 \text{ nm}^2$, using the scripted symmetry option of D+. The deviations between the 3 models were small. The larger finite sized slabs were somewhat closer to the finite model of X+. Fig. S15 shows an equivalent comparison, using the asymmetric slab model of D+ instead of the symmetric slab model. Both models adequately fit to the results of X+.

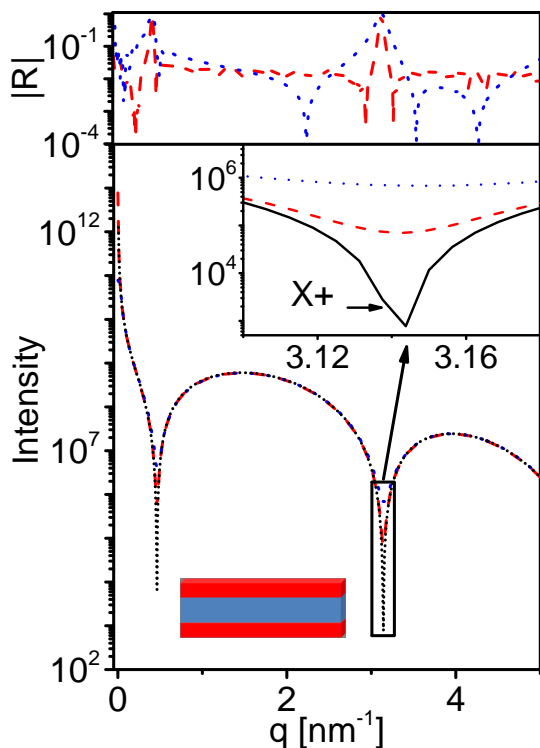


Fig. S14. Modeled X-ray scattering intensity from layers of symmetric slabs. Each slab has an inner layer that is 2 nm thick with an electron density of 280 e/nm^3 that is sandwiched between two outer slabs that are each 1 nm thick and their electron density is 400 e/nm^3 . The solvent electron density is 334 e/nm^3 . (see cartoon). The black solid curve was computed, using an infinite slab in X+. The broken red curve was computed in D+, using $1,000 \times 1,000 \text{ nm}^2$ slab that was constructed from 10,000 slabs of $10 \times 10 \text{ nm}^2$ using Scripted Symmetry and Grid Size of 100. Orientation average was calculated using adaptive (VEGAS) Monte Carlo integration. The broken blue curve was computed on CPU with the Direct method in D+, using a $100 \times 100 \text{ nm}^2$ slab. Orientation average was calculated using adaptive Gauss Kronrod integration. The curves at the top show the normalized absolute residuals, $|R|$, where the reference curve is the intensity computed by X+ (Eq. S2).

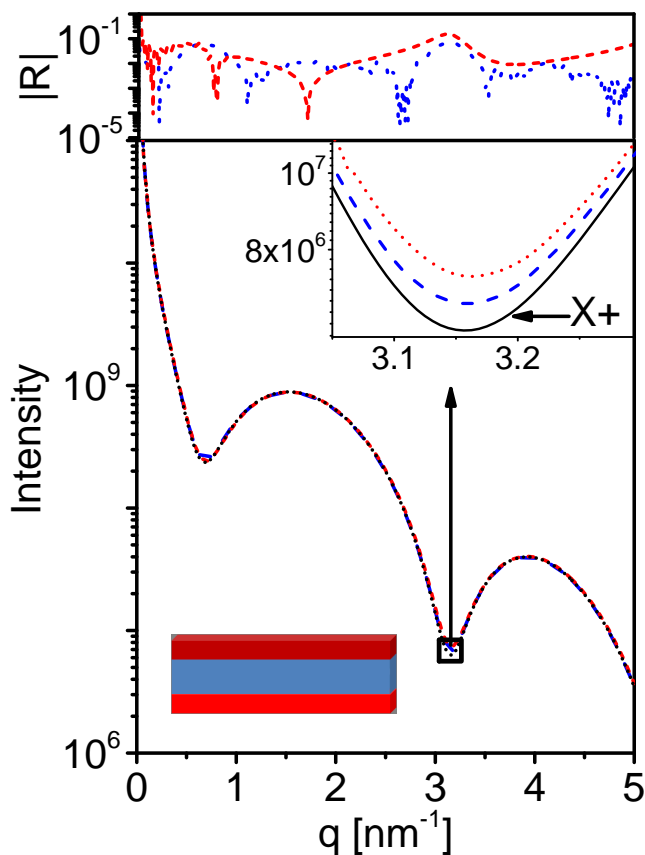


Fig. S15. Modeled X-ray scattering intensity from layers of asymmetric slabs. Each slab has an inner layer that is 2 nm thick with an electron density of $280 \text{ e}/\text{nm}^3$, which is sandwiched between two outer slabs, each 1 nm thick. One of the outer slabs has an electron density of $400 \text{ e}/\text{nm}^3$ and the other has an electron density of $450 \text{ e}/\text{nm}^3$ (see cartoon). The black curve was computed, using an infinite slab in X+. The broken blue curve was computed in D+, using $1,000 \times 1,000 \text{ nm}^2$ slab that was constructed from 10,000 slabs of $10 \times 10 \text{ nm}^2$ using Scripted Symmetry and Grid Size of 100. Orientation average was calculated using adaptive (VEGAS) Monte Carlo integration. The broken red curve was computed on CPU with the Direct method in D+, using a $100 \times 100 \text{ nm}^2$ slab. Orientation average was calculated using adaptive Gauss Kronrod integration. The top curve shows the normalized absolute residual, $|R|$, where the reference curve is the intensity computed by X+ (Eq. S2).

Fig. S16 compares between $100 \times 100 \text{ nm}^2$ symmetric slab models within D+, computed in different ways. The Direct method was computed on the CPU and compared with several GPU computations. The GPU computations were done with either a single $100 \times 100 \text{ nm}^2$ slab model or with 100 identical units of $10 \times 10 \text{ nm}^2$ slabs. The

Hybrid method with either Space-Filling symmetry, or Scripted Symmetry, or Manual Symmetry were used to compute the 100 units. All the methods gave very similar results.

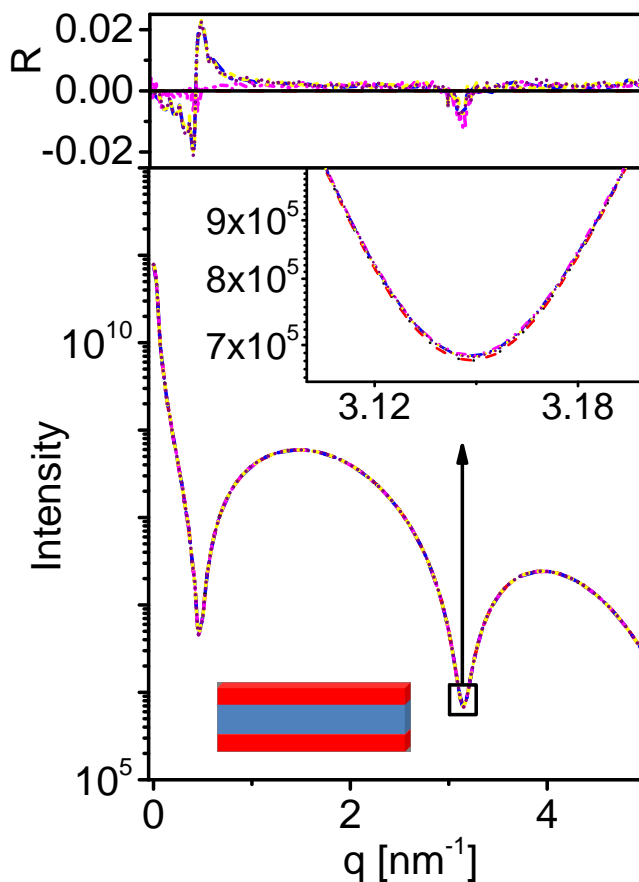


Fig. S16. Modeled X-ray scattering intensity from layers of symmetric slabs, which are isotropically distributed in random orientations in solution. Each slab has an inner layer that is 2 nm thick with an electron density of $280 \text{ e}/\text{nm}^3$ that is sandwiched between two outer slabs that are each 1 nm thick and their electron density is $400 \text{ e}/\text{nm}^3$. The solvent electron density is $334 \text{ e}/\text{nm}^3$ (see cartoon). All curves were computed in D+, using $100 \times 100 \text{ nm}^2$ slabs with different methods. Direct computation from Fig. S14 (black) is compared to a Direct computation on CPU using the asymmetric slab model (red). Several computations were done using GPU as follows. A single $100 \times 100 \text{ nm}^2$ slab was computed with a **Grid Size** of 460 and orientation average was computed using adaptive (VEGAS) Monte Carlo integration (magenta). Alternatively, the Hybrid method was used to compute the scattering of the same structure from 100 copies of $10 \times 10 \text{ nm}^2$ slabs, for which a **Grid Size** of 100 was used. To create the large slab, Scripted Symmetry (purple), Space-Filling Symmetry (yellow), and Manual Symmetry (blue) were used. The top curve shows the normalized residual, R , where the reference curve is the intensity computed by D+ on the CPU, using the Direct method (Eq. S2).

Fig. S17 compares between a stack of 10 infinite multi-layered slabs, computed by X+, with a model of a stack of 10 finite slabs (or rectangular boxes) models that formed a similar stack of large uniform slabs, computed by D+. The results are adequately similar, considering the fact that D+ computes finite size slabs and X+ computes infinite size slabs.

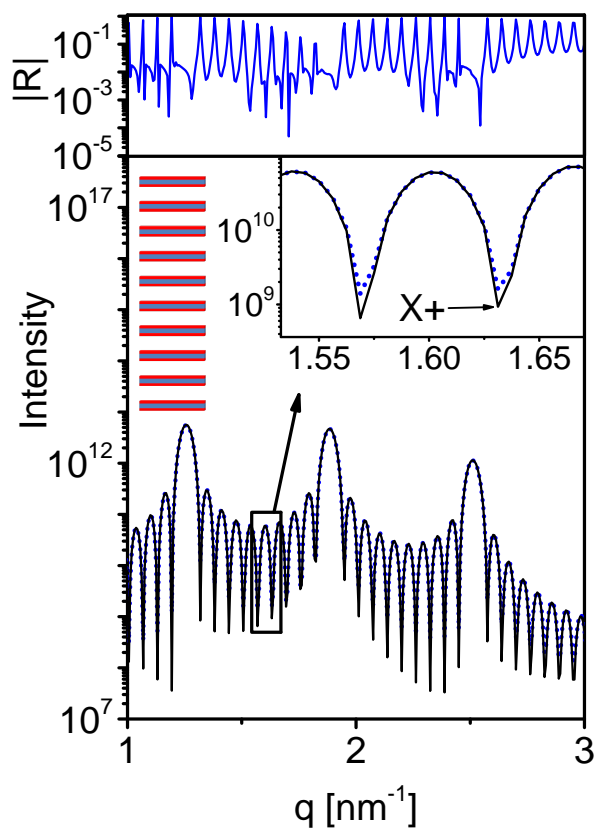


Fig. S17. Modeled X-ray scattering intensity from Multi-layered structure containing 10 layers of symmetric slabs, which are isotropically distributed in random orientations in solution. Each slab has an inner layer that is 2 nm thick with an electron density of 280 e/nm^3 that is sandwiched between two outer slabs that are each 1 nm thick and their electron density is 400 e/nm^3 . Between adjacent layers there is a 6 nm water gap with an electron density of 334 e/nm^3 (see cartoon). The solid black curve was computed, using a stack of infinite slabs in X+. The broken (blue) curve was computed in D+, using $1,000 \times 1,000 \text{ nm}^2$ slabs. Each slab was constructed from 10,000 slabs of $10 \times 10 \text{ nm}^2$ using Scripted Symmetry and Grid Size of 100. Orientation average was calculated using adaptive (VEGAS) Monte Carlo integration. The top curve shows the normalized absolute residual, $|R|$, where the reference curve is the intensity computed by X+ (Eq. S2).

9.2. Atomic models

We compared the scattering curves from atomic models of D+ with those of CRY SOL (ATSAS 2.8.2) (Svergun *et al.*, 1995), and *DebyeCalculator*. *DebyeCalculator* uses the equations of D+ but instead of numerical orientation average it uses Debye formula (Debye, 1915), which applies analytic orientation averaging. Debye formula is applicable for spherically-symmetric objects like atoms.

MOLProbity (Chen *et al.*, 2010) was used to add hydrogen atoms to PDB files. PyMOL (Schrödinger, LLC, 2015) was used to add charges to residues that were supposed to be charged but the PDB file did not include the charges. The atomic form factor of ions, like O^- , differ from the form-factor of the corresponding neutral atom (O, in this example). Both D+ and *DebyeCalculator* take into account the atomic form factor of the 209 atoms and ions listed in the International Tables for Crystallography, using the five-Gaussian approximation (Hamilton, 1974; Marsh & Slagle, 1983). N^+ , however, is not included in the International Tables for Crystallography and both D+ and *DebyeCalculator* computes the form factor of an N atom instead. If both hydrogen atoms and charges are added to an N^+ group, an excess electron is obtained. An approximation that might be used to better account for an N^+ ion, is to manually remove from the PDB file, one of the hydrogen atoms next to the N^+ ion. This ensures that the correct total number of electrons in the protein is preserved.

CRY SOL does not take the contribution from any ion into account. Hence, to properly compare between CRY SOL and D+ (or *DebyeCalculator*), charges were either not added or removed from the PDB files. Computations in CRY SOL were done using the parameters that provide maximum accuracy. Hence, 99 was the maximum order of harmonics, and 18 was the order of Fibonacci grid.

In all the computations in D+, 0.0001 was the **Convergence** tolerance, the solvent electron density was $334 e/nm^3$, and the type of solvent radius was **Dummy Atoms**. Unless

otherwise is indicated, Monte Carlo (Mersenne Twister) was the orientation averaging integration method.

9.2.1. Tubulin Dimer Tubulin heterodimer is made of α -tubulin and β -tubulin, which are homologous and form the so called tubulin dimer. Depending on solution conditions, tubulin dimer may assemble into MT and/or tubulin rings (Diaz *et al.*, 1994). Fig. S18 compares between the scattering curve from the central tubulin dimer in PDB ID 3J6F (which contains nine dimers), by *DebyeCalculator*, CRY SOL, and D+. H atoms were explicitly added to the PDB file (Chen *et al.*, 2010). Fig. S19 shows a similar comparison, however, H atoms were only implicitly taken into account. Figs. S20 and S21 repeats the same comparison, but without removing the charges from the PDB file.

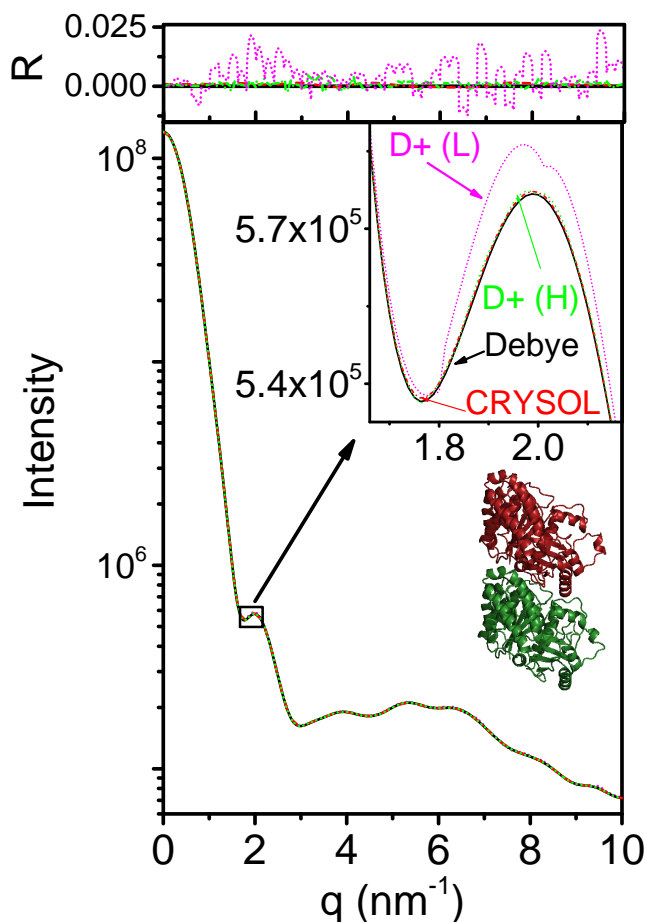


Fig. S18. Modeled X-ray scattering intensity from an atomic structure of tubulin dimer (PDB ID 3J6F), shown as a ribbon diagram. The dimers were assumed to adopt isotropic distributed in random orientations in solution. H atoms were explicitly added to the PDB file (Chen *et al.*, 2010). The solid black curve was computed by *DebyeCalculator*. The red broken curve was computed by CRY SOL and the broken pink and green curves were computed by D+ at low (L, pink broken curve) and high (H, green broken curve) resolution. **Grid Size** of 100 and **Convergence** tolerance of 10^{-4} were used for low resolution computations, and **Grid Size** of 250 and **Convergence** tolerance of 10^{-7} were used for high resolution. The inset shows the same computations on an expanded scale. The top curve shows the normalized residual, R , where the reference curve is the intensity computed by *DebyeCalculator* (Eq. S2).

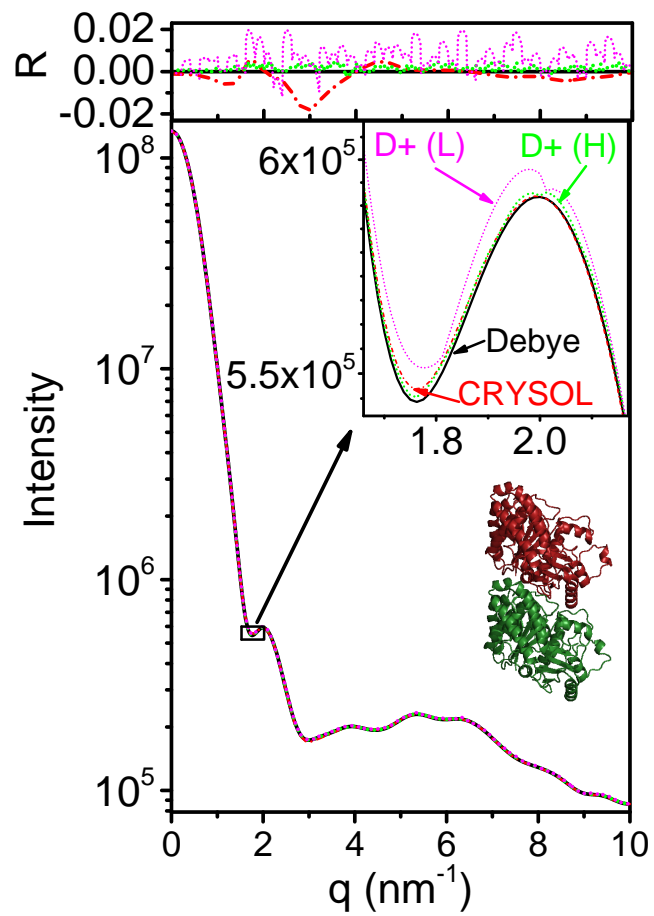


Fig. S19. Modeled X-ray scattering intensity as in Fig. S18 but H atoms were implicitly (rather than explicitly) taken into account.

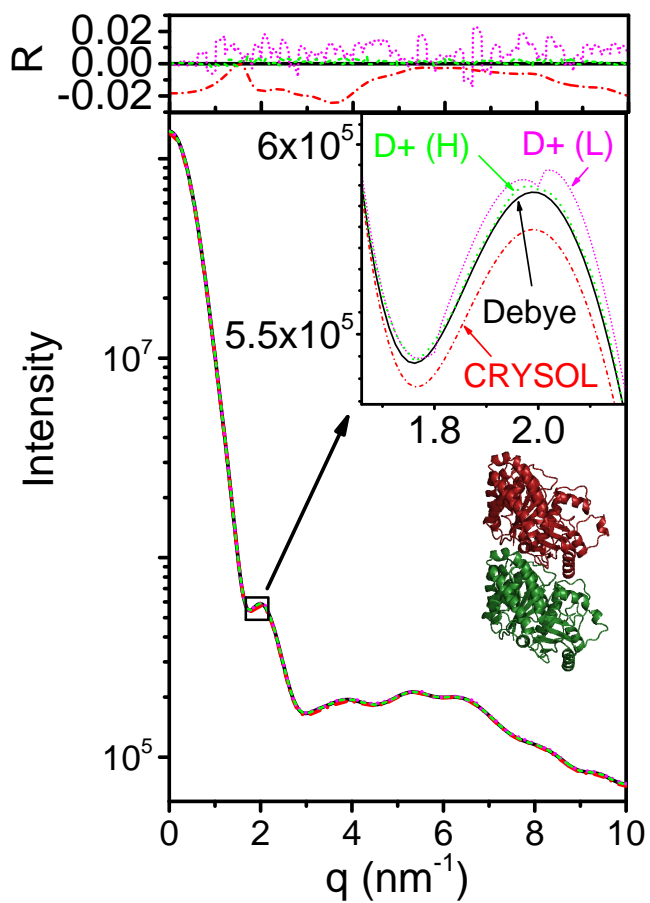


Fig. S20. Modeled X-ray scattering intensity as in Fig. S18 but charges (on charged residues) were added using PyMOL (Schrodinger, LLC, 2015).

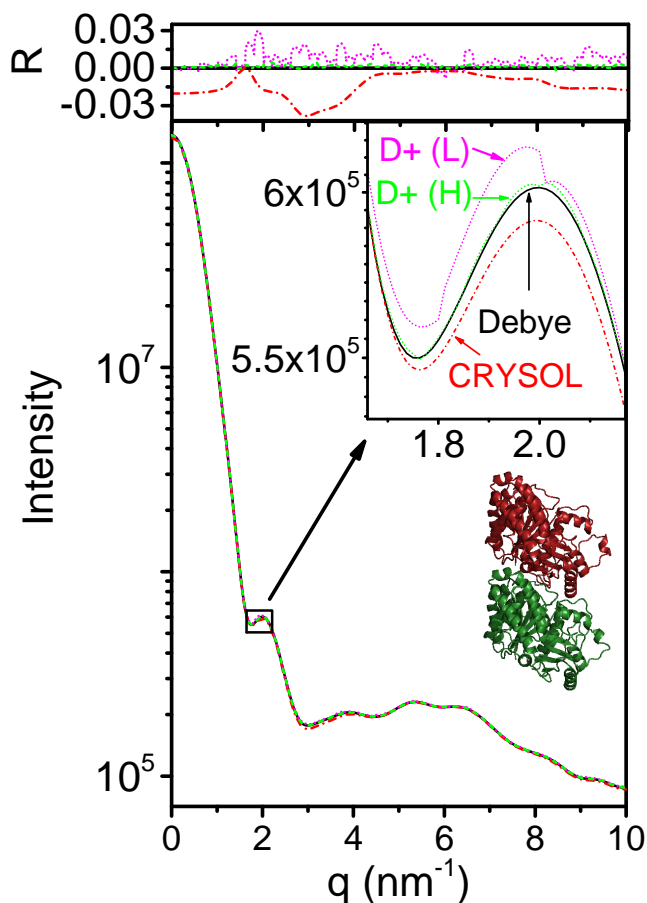


Fig. S21. Modeled X-ray scattering intensity as in Fig. S19 but charges (on charged residues) were added using PyMOL (Schrödinger, LLC, 2015).

Fig. S22 compares between different computation methods within D+. Tubulin dimers, with explicitly added H atoms and after removing the charges, were computed in different ways in D+. The computations are compared with *DebyeCalculator*, whose results were comparable with the results of CRY SOL (Fig. S18). The computations of D+ were either done on CPU or GPU, using the Direct method (implemented only for CPU) and the reciprocal grid (RG) algorithm. When using the RG algorithm, different orientation average methods were used: Adaptive Gauss-Kronrod (using CPU), adaptive (VEGAS) Monte Carlo (using GPU), or regular Monte Carlo (using CPU or GPU).

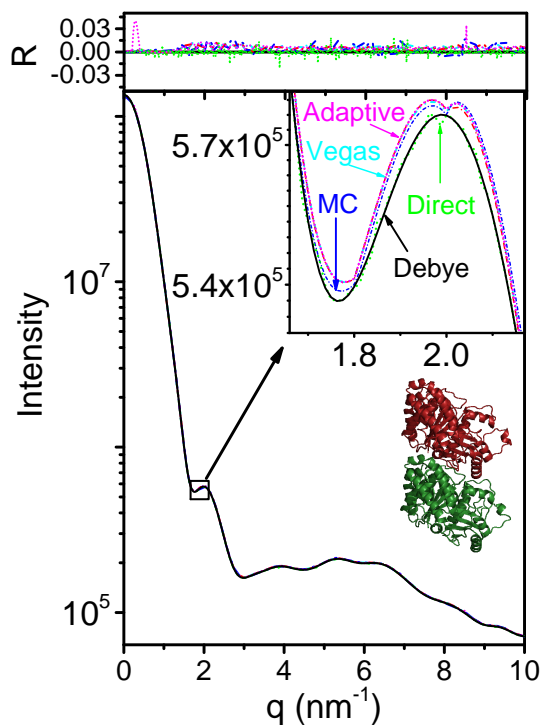


Fig. S22. Modeled X-ray scattering intensity from an atomic model of tubulin dimer (PDB ID 3J6F), shown as a ribbon diagram. The dimers were assumed to adopt isotropic distributed in random orientations in solution. H atoms were explicitly added to the PDB file (Chen *et al.*, 2010). The solid black curve was computed by *DebyeCalculator*, which applies Debye formula (Debye, 1915). The broken green curve was computed by D+ using the Direct method (on CPU). The other broken curves were computed by the reciprocal grid (RG) algorithm of D+, where orientation average was computed by the following integration algorithms: Adaptive Gauss-Kronrod (magenta - using CPU only, purple - using GPU for computing the grid and CPU for orientation average), adaptive (VEGAS) Monte Carlo (cyan, using GPU), or Monte Carlo (blue, using CPU, red using GPU). The inset shows a small part of the curve on an expanded scale. The top curves show the normalized residual, R , where the reference curve is the intensity computed by *DebyeCalculator* (Eq. S2).

9.2.2. Tubulin ring Under certain experimental conditions (high tubulin concentration and low temperature), tubulin dimers may assemble into rings (Diaz *et al.*, 1994). To get the ring structure the dimer subunit should be docked into an assembly symmetry, which involves both rotations and translations. To compute the scattering curve from a tubulin ring, we used the same central tubulin dimer from PDB ID 3J6F and explicitly added H atoms. We then rotated the dimer by $\gamma = 180^\circ$ (about the z -axis), $\alpha = 90^\circ$ (about the x -axis), and $\beta = 50^\circ$ (about the y -axis). This rotation puts the dimer in a parallel direction to the tangent of the ring. The ring itself was built from 13 dimers that were rotated about the z axis by $\gamma = \frac{360^\circ}{13}n$ (in the plane of the ring), and translated by $x = R_{Ring} \sin\left(\frac{360^\circ}{13}n\right)$ and $y = R_{Ring} \cos\left(\frac{360^\circ}{13}n\right)$ (where $n \in [0, 1, \dots, 12]$). A PDB file of the ring (containing all the 13 dimers) was then created and its scattering curve was computed by CRY SOL and by *DebyeCalculator*. The results were compared with the computations of D+ using the PDB file of the rotated dimer and the Hybrid method to account for the assembly symmetry of the ring. Grid was computed only for the dimer. The computations are compared in Fig. S23, using implicit H atoms, and in Fig. S24, using explicit H atoms.

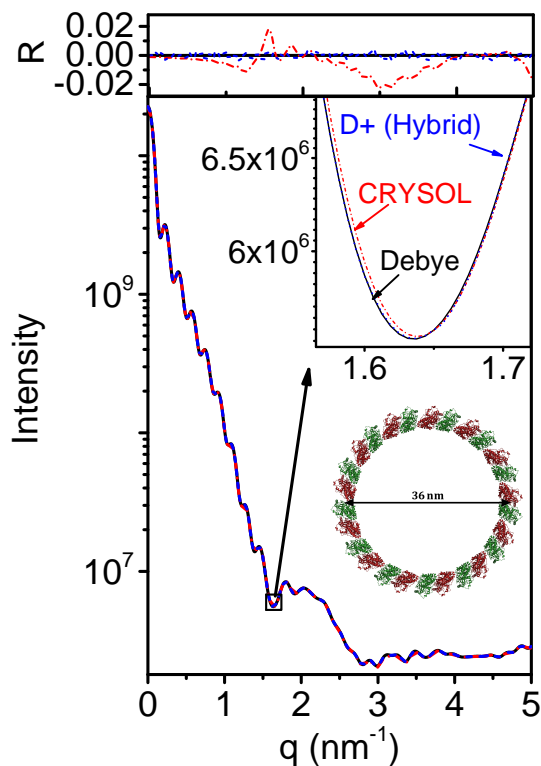


Fig. S23. Modeled X-ray scattering intensity from an atomic structure of a tubulin ring, shown as a ribbon diagram. The rings were assumed to adopt isotropic distributed in random orientations in solution. H atoms were implicitly taken into account (PDB ID 3J6F). The solid black curve was computed by *DebyeCalculator*, which applies Debye formula (Debye, 1915). The broken red curve was computed by CRY SOL. Both used a single PDB file for the entire ring. The broken blue curve was computed by D+ on a GPU, using a `Grid Size` of 200 for the tubulin dimer, and the Hybrid method for the ring symmetry. Orientation average was computed using Monte Carlo integration with `Convergence` tolerance of 10^{-4} . The inset shows a small part of the curve on an expanded scale. The top curves show the normalized residuals, R , where the reference curve is the intensity computed by *DebyeCalculator* (Eq. S2).

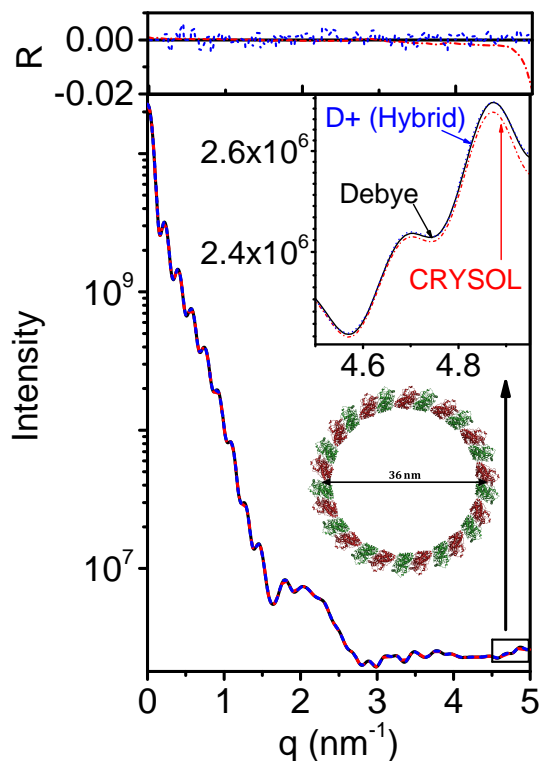


Fig. S24. Modeled X-ray scattering intensity from an atomic structure of tubulin ring, shown as a ribbon diagram. The rings were assumed to adopt isotropic distributed in random orientations in solution. H atoms were explicitly added to the PDB file of the dimer (Chen *et al.*, 2010) (PDB ID 3J6F). The solid black curve was computed by *DebyeCalculator*, which applies Debye formula (Debye, 1915). The broken red curve was computed by CRY SOL. The broken blue curve was computed by D+ on GPU, using a **Grid Size** of 200 for the tubulin dimer, and the Hybrid method for the ring symmetry. Orientation average was computed using Monte Carlo integration with **Convergence** tolerance of 10^{-4} . The inset shows a small part of the curve on an expanded scale. The top curves show the normalized residual, R , where the reference curve is the intensity computed by *DebyeCalculator* (Eq. S2).

We then compared different ways to compute the scattering from tubulin ring, within D+. The results are shown in Fig. S25 (for explicit H atoms) and in Fig. S26 (for implicit H atoms). Computations were done by using a single PDB file for the entire ring by either D+ or *DebyeCalculator*, or in D+ using the grid of the dimer and the Hybrid method for the ring symmetry, or using the grid for the entire ring symmetry, or using the Direct method.

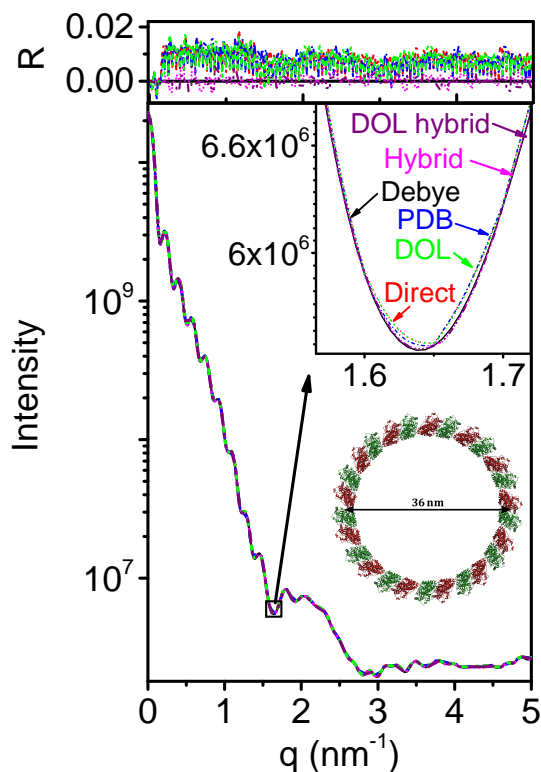


Fig. S25. Modeled X-ray scattering intensity from an atomic structure of tubulin ring, shown as a ribbon diagram. The rings were assumed to adopt isotropic distributed in random orientations in solution. H atoms were explicitly added to the PDB file of the dimer (PDB ID 3J6F) (Chen *et al.*, 2010). Unless otherwise indicated, computations were done on GPU. The solid black curve was computed by *DebyeCalculator*, which applies Debye formula (Debye, 1915). The broken red curve was computed by D+ on the CPU, using the Direct method. The broken blue curve was computed by D+ using a single PDB file for the entire ring (which was used for CRY SOL and *DebyeCalculator*). The broken magenta curve was computed using a grid of tubulin dimer (**Grid Size** of 200) in a Manual Symmetry that was computed using the Hybrid method. The broken green curve was computed by the dimer in Manual Symmetry using a **Grid Size** of 200 for both the tubulin and the ring. The broken purple curve was computed using the grid of the dimer (**Grid Size** of 200) in a Scripted Symmetry, which was computed by the Hybrid method. The inset shows a small part of the curves on an expanded scale. The top curves show the normalized residuals, R , where the reference curve is the intensity computed by *DebyeCalculator* (Eq. S2).

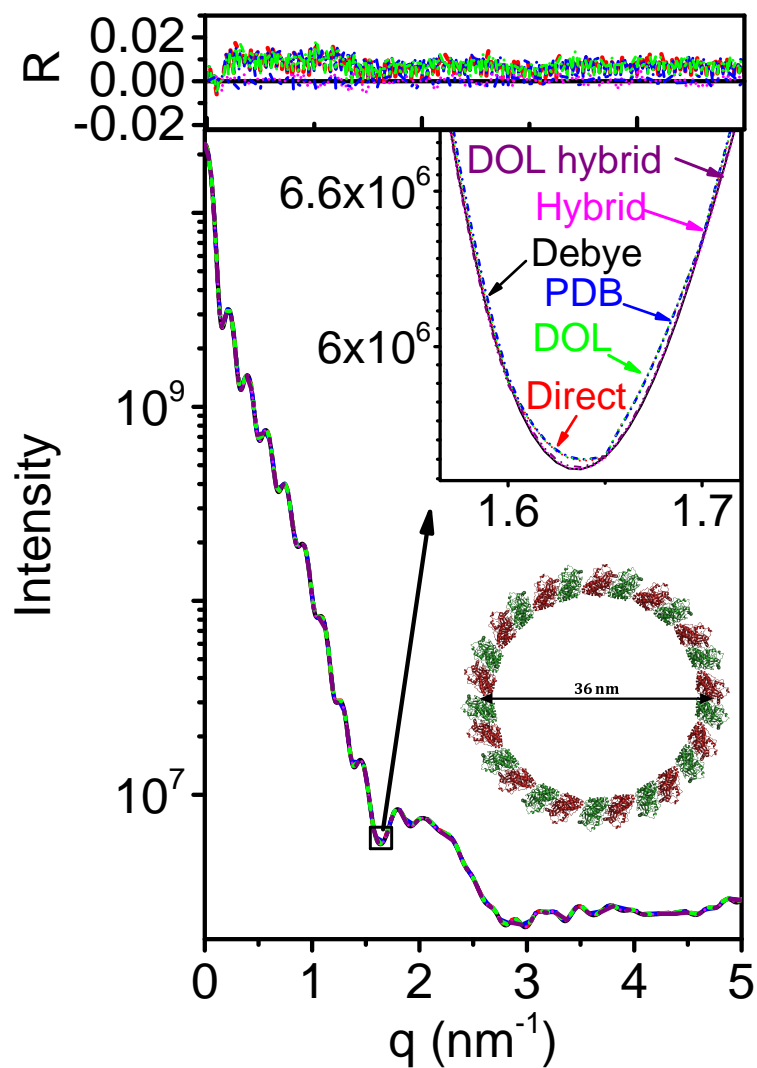


Fig. S26. Modeled X-ray scattering intensity from an atomic structure of tubulin ring, shown as a ribbon diagram. The rings were assumed to adopt isotropic distributed in random orientations in solution. H atoms were implicitly taken into account. Unless otherwise indicated, computations were done on GPU. The solid black curve was computed by *DebyeCalculator*, which applies Debye formula (Debye, 1915). The broken red curve was computed by D+ on the CPU, using the Direct method. The broken blue curve was computed by D+ using a single PDB file for the entire ring. The broken magenta curve was computed from a grid of tubulin dimer (PDB ID 3J6F) in a Manual Symmetry, using the Hybrid method. The broken green curve was computed from the dimer in a Manual Symmetry, using a grid for the ring. The broken purple curve was computed from the grid of the dimer in a Scripted Symmetry, using the Hybrid method. The size of the grid was fixed at 200. The inset shows a small part of the curve on an expanded scale. The top curves show the normalized residual, R , where the reference curve is the intensity computed by *DebyeCalculator* (Eq. S2).

9.2.3. Tubulin is a Space-Filling symmetry To test the option of Space-Filling symmetry with atomic models, we used the same central tubulin dimer from PDB ID 3J6F. No H atoms or charges were added to the PDB file. The dimer was docked into a simple cubic Space-Filling symmetry (Fig. S27A). The scattering curve was computed by D+, using different options for defining the assembly symmetry (Space-Filling, Scripted, or Manual), using Grids for the entire structure or the Hybrid method where a Grid was computed for the dimer, or by loading the precomputed scattering amplitude from the dimer and putting it in the assembly symmetry. Each computation was repeated with different orientation average integration method: Monte Carlo, adaptive (VEGAS) Monte Carlo, or adaptive Gauss Kronrod. A PDB file of the entire cubic structure (containing 18 dimers) was then created and its scattering curve was computed by *DebyeCalculator*. Fig. S27A shows that all the computations of D+ resulted in the same curve, which agreed with the curve obtained by *DebyeCalculator*. Fig. S27B shows that *DebyeCalculator* is in good agreement with CRY SOL. When solvation layer was added to the PDB file of the entire structure (Fig. S27C) or to the dimer (Fig. S27D), all the symmetry generation and integration methods led to similar results.

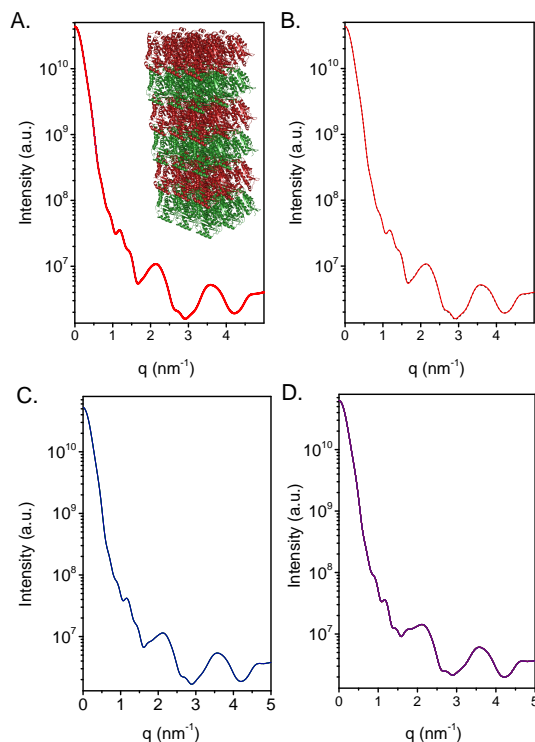


Fig. S27. Modeled X-ray scattering intensity from an atomic structure of tubulin in a simple cubic Space-Filling symmetry, shown as a ribbon diagram. The cubic lattices were assumed to adopt isotropic distributed in random orientations in solution. The central tubulin dimer from PDB ID 3J6F was rotated by $\gamma = 180^\circ$ (about the z -axis), and $\alpha = 180^\circ$ (about the x -axis). H atoms were implicitly taken into account. The PDB was then docked into a simple cubic Space-Filling symmetry with 3 subunits in the x direction with a repeat distance of 4 nm, 2 subunits in the y direction with a repeat distance of 4 nm, and 3 subunits in the z direction with a repeat distance of 8.13 nm. **A.** Scattering curves were computed in D+ using different options to define the assembly symmetry (Space-Filling, Scripted Symmetry, or Manual Symmetry), using Grids, or the Hybrid method (with a Grid for the dimer), or by loading the precomputed scattering amplitude from the dimer. Each computation was repeated using different orientation average integration methods: Monte Carlo, adaptive (VEGAS) Monte Carlo, or Gauss Kronrod. A PDB file of the entire structure (containing all the 18 dimers) was then created and its scattering curve was computed by *DebyeCalculator*. **B.** The curve that was computed by *DebyeCalculator* in **A** (solid red curve) is compared with the curve computed by CRYSOLO for the same PDB (broken curve). **C.** The PDB of the entire structure was computed with a solvation layer whose thickness was 3 nm and electron density $364 \text{ e}\cdot\text{nm}^{-3}$. The computation was done by computing Grids or by loading a pre-computed scattering amplitude. Orientation average was done using Monte Carlo, adaptive (VEGAS) Monte Carlo, or Gauss Kronrod integrations. **D.** The computation from **A** were repeated, using solvated tubulin dimer. The solvation layer was 3 nm thick and its electron density was $364 \text{ e}\cdot\text{nm}^{-3}$.

9.3. Atomic vs. geometric models

We created PDB files in which oxygen atoms were randomly close packed into geometric shapes. We used the PDB files to compare their scattering curve with the curves of their corresponding geometric models. All the curves were computed in D+. The PDB files were computed using the `Solvent Only` option in D+, where the solvent method was `Dummy Atoms Voxelized`, which gave the voxels forming the enveloping geometric shape (the `Solvent Only` option did not compute the form-factors of the oxygen atoms).

Fig. S28 compares atomic with geometric spherical models. Fig. S29 compares a solvated atomic spherical model with the equivalent geometric core-shell spherical model, where the shell corresponds to the solvation layer. We then compare between atomic and geometric models of a hollow sphere (Fig. S31) and then the corresponding solvated hollow sphere models (Fig. S31), in which solvation layers are both internal and external. The comparisons validate the solvation algorithm used in D+. We then tested the cylindrical (Fig. S32) and the slab (Fig. S33) geometries. Within the tested q -range (between 0 and 5 nm^{-1}), all the atomic and their corresponding geometric models were very similar (in other words, the residuals were very small).

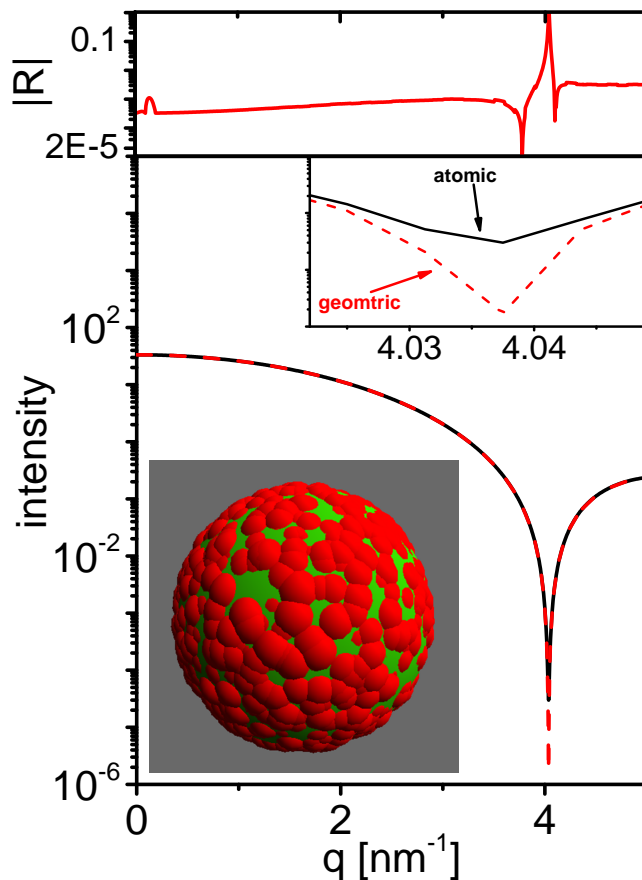


Fig. S28. Modeled X-ray scattering intensity of atomic and geometric spherical structures. The structures were assumed to adopt isotropic distributed in solution. The solid black curve was computed from a PDB file, in which oxygen atoms were randomly close packed into a sphere. The computation was done in D+ from the atomic model using Solvent Only, Dummy Atoms Voxelized, solvent electron density of $1 \text{ e}/\text{nm}^3$, and no solvation shell. The broken (red) curve was computed in D+, using the corresponding geometric sphere model with a radius of 1.113 nm and an electron density of $1 \text{ e}/\text{nm}^3$. The solvent electron density was set to 0, corresponding to vacuum. The red curve at the top inset shows the normalized absolute residuals, $|R|$, as a function of the magnitude of the scattering vector, q , where the reference curve is the intensity of the atomic model (Eq. S2). The cartoon shows the closed packed spherical atomic model in Red and the corresponding geometric model in green.

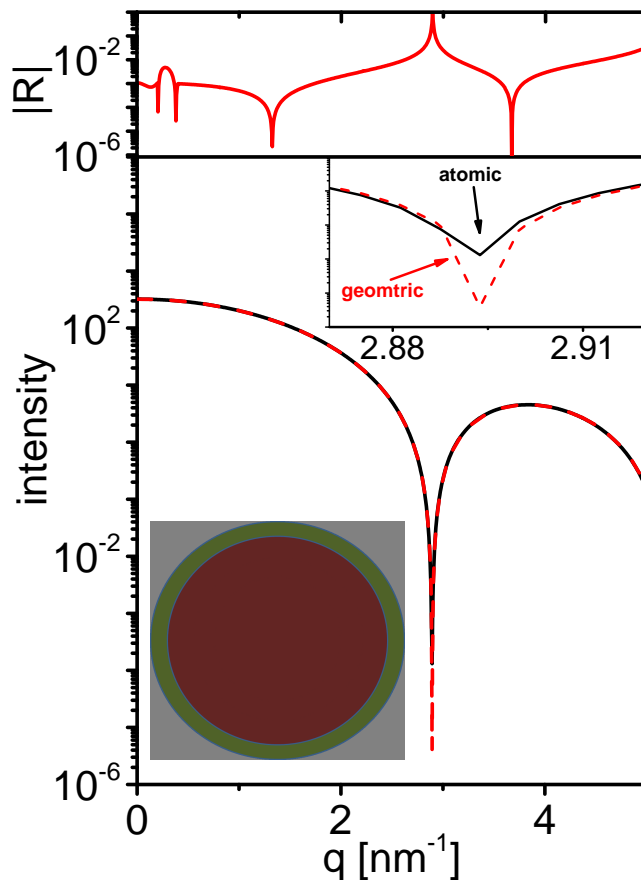


Fig. S29. Modeled X-ray scattering intensity of solvated atomic spherical model and a geometric model of a spheres and a thin shell. The solid black curve was computed from a PDB file, in which oxygen atoms were randomly close packed into a sphere. The computation was done in D+ from the atomic model using Solvent Only, Dummy Atoms Voxalized, solvent electron density $1 \text{ e}/\text{nm}^3$, an outer solvent electron density of $-1 \text{ e}/\text{nm}^3$, and solvation thickness of 0.30 nm . These parameters produced a second shell with an electron density contrast of $+1$ to the core. The broken (red) curve was computed in D+, using the corresponding geometric core-shell spherical model, with a core radius of 1.113 nm and an electron density of $1 \text{ e}/\text{nm}^3$, and an outer shell with a thickness of 0.30 nm and an electron density of $2 \text{ e}/\text{nm}^3$. The top curve shows the normalized absolute residual, $|R|$, as a function of q , where the reference curve is the intensity computed from the atomic model (Eq. S2). The cartoon shows a cross section of the geometric model.

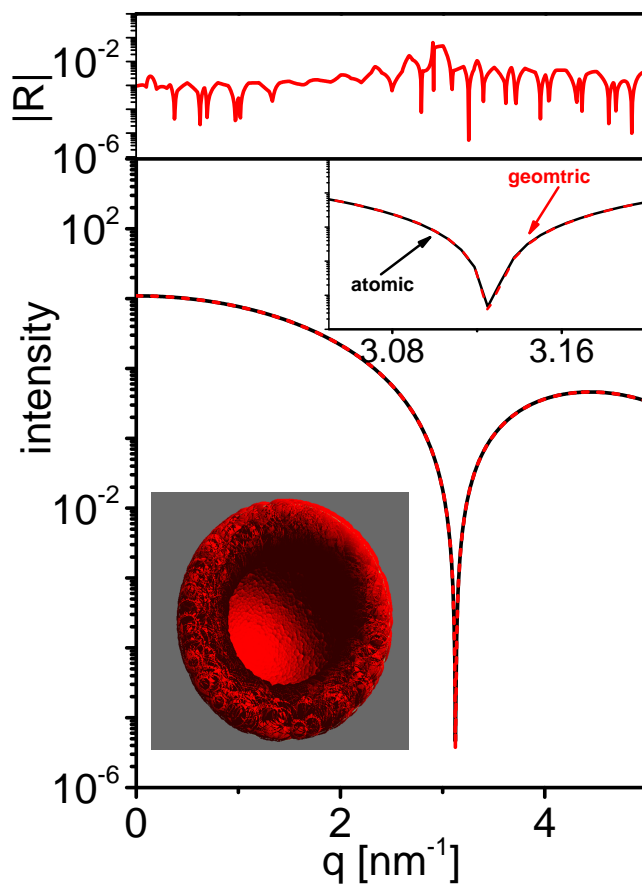


Fig. S30. Computed X-ray scattering intensity of an atomic model of a hollow sphere and a geometric model of a hollow sphere. The structures were assumed to adopt isotropic distributed in solution. The solid black curve was computed from a PDB file, in which oxygen atoms were randomly close packed into a hollow sphere. The computation was done in D+ from the atomic model using Solvent Only, Dummy Atoms Voxelized, solvent electron density 1 e/nm³ and no solvation shell. The broken (red) curve was computed in D+, using the corresponding geometric hollow sphere model with an inner radius of 0.868 nm and an outer radius of 1.130 nm, and an electron density of 1 e/nm³. The solvent electron density was set to 0, corresponding to vacuum. The top curve shows the normalized absolute residual, $|R|$, as a function of q , where the reference curve is the intensity computed from the atomic model (Eq. S2). The cartoon shows a cross section of the atomic model in red.

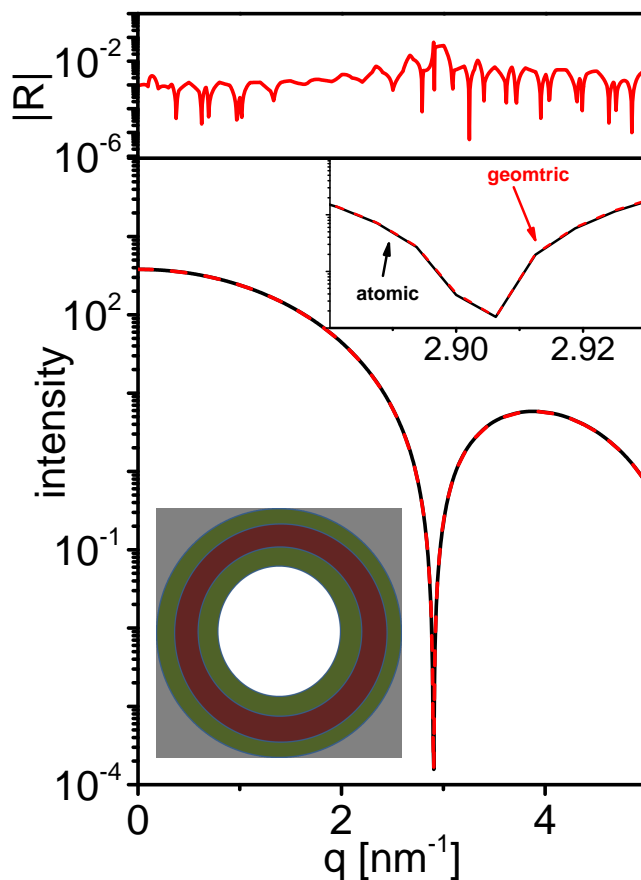


Fig. S31. Computed X-ray scattering intensity from a solvated hollow sphere atomic model and a geometric concentric multi layered spherical shell model. The solid black curve was computed using the atomic model from Fig. S30 with Solvent Only, Dummy Atoms Voxelized, solvent electron density $1 \text{ e}/\text{nm}^3$, solvation thickness 0.3 nm , and an outer solvent electron density of $-1 \text{ e}/\text{nm}^3$. The **Fill Holes** check-box remained unchecked. The broken (red) curve was computed in D+, using the corresponding concentric multi layered spherical shell model. The first spherical shell is 0.301 nm thick, its inner radius is 0.575 nm , and its electron density is $2 \text{ e}/\text{nm}^3$. The second spherical shell is 0.256 nm thick, and its electron density is $1 \text{ e}/\text{nm}^3$. The third spherical shell is 0.295 nm thick, and its electron density is $2 \text{ e}/\text{nm}^3$. The top curve shows the normalized absolute residual, $|R|$, as a function of q , where the reference curve is the intensity computed from the atomic model (Eq. S2). The cartoon shows a cross section of the geometric model.

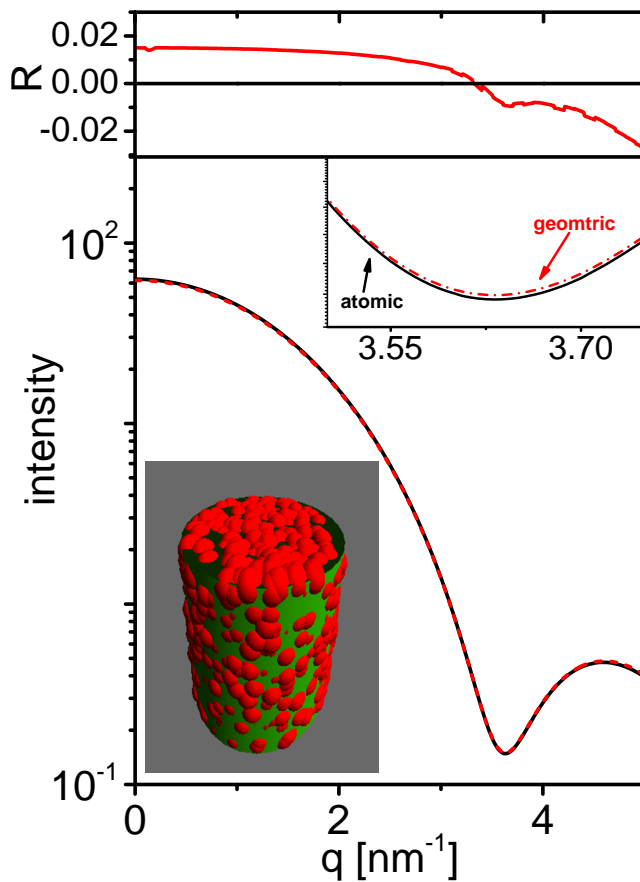


Fig. S32. Computed X-ray scattering intensity from an atomic rod model and a geometric model of a uniform rod. The structures were assumed to adopt isotropic distributed in random orientations in solution. The solid black curve was computed from a PDB file, in which oxygen atoms were randomly close packed into a rod. The computation was done in D+ from the atomic model with Solvent Only, Dummy Atoms Voxelized, solvent electron density $1 \text{ e}/\text{nm}^3$, and no solvation. The broken (red) curve was computed in D+, using the corresponding rod model, with a rod radius of 1.088 nm , rod height of 2.121 nm , and an electron density of $1 \text{ e}/\text{nm}^3$. The solvent electron density was set to 0, corresponding to vacuum. The top curve shows the normalized absolute residual, $|R|$, as a function of q , where the reference curve is the intensity computed from the atomic model (Eq. S2). The cartoon shows that atomic model in Red and the corresponding geometric model in green.

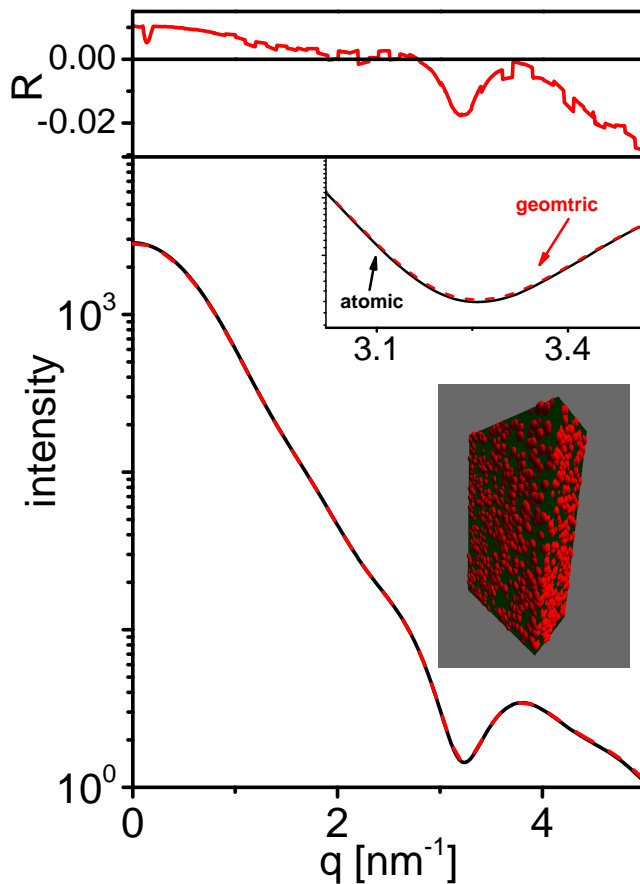


Fig. S33. Computed X-ray scattering intensity from atomic and geometric models of a uniform rectangular slab. The structures were assumed to adopt isotropic distributed in random orientations in solution. The solid black curve was computed from a PDB file, in which oxygen atoms were randomly close packed into a rectangular slab. The computation was done in D+ from the atomic model with Solvent Only, Dummy Atoms Voxelized, solvent electron density $1 \text{ e}/\text{nm}^3$, and no solvation. The broken (red) curve was computed in D+, using the corresponding uniform slab model with an electron density of $1 \text{ e}/\text{nm}^3$ and dimensions of $3.055 \text{ nm} \times 2.115 \text{ nm} \times 4.108 \text{ nm}$. The solvent electron density was set to 0, corresponding to vacuum. The top curve shows the normalized residual, $|R|$, as a function of q , where the reference curve is the intensity computed from the atomic model (Eq. S2). The cartoon shows the atomic model in red and the geometric model in green.

9.4. Fitting algorithm tests

The fitting algorithm of D+ was tested against calculated (Figs. S34,S35,S36), and experimental scattering curves of lipid membranes (Fig. S37), or protein solutions

(Figs. S38 and S39). A scattering signal, with known parameters, served as the target signal, to which D+ was trying to fit its computed curve. The mutable model parameters were described in the figure captions with the initial values (in parenthesis) and the final fitted values (after the parenthesis). The figures show adequate fits.

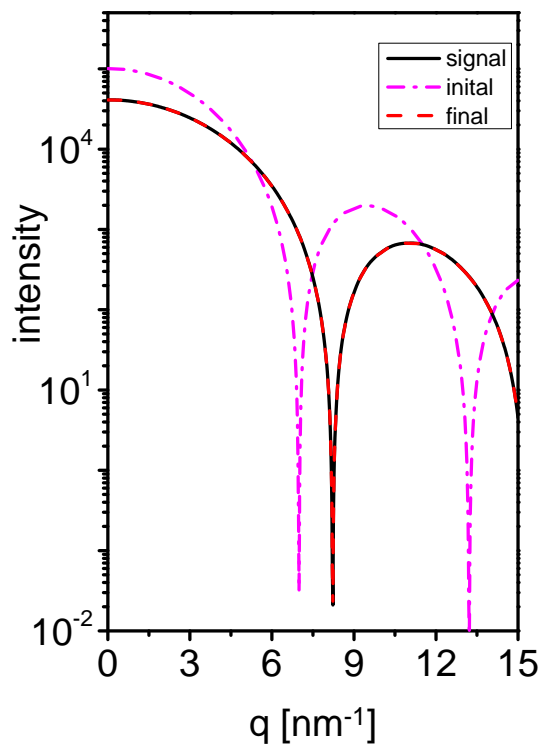


Fig. S34. Fitting to a modeled X-ray scattering intensity curve that was computed from a structure of a uniform sphere surrounded by four concentric uniform hollow spherical shells. The structures were assumed to adopt isotropic distributed in solution. The radius of the internal sphere is (0.15)0.1 nm and its electron density is 400 e/nm^3 . The first spherical shell is (0.15)0.1 nm thick, and its electron density is 500 e/nm^3 . The second spherical shell is (0.05)0.1 nm thick, and its electron density is 600 e/nm^3 . The third spherical shell is (0.08)0.1 nm thick, and its electron density is 700 e/nm^3 . The fourth spherical shell is (0.15)0.1 nm thick, and its electron density is 800 e/nm^3 . The solid black curve was computed in D+, and used as the target signal for the fitting. The broken magenta curve was computed by D+, using the initial parameters (given in the parenthesis, if mutable) and the broken red curve is the result of the fit. Fitting was done using Levenberg-Marquardt algorithm, Ratio Residuals, Trivial Loss, with 20 iterations. Step Size was 10^{-5} , the Convergence tolerance was 10^{-6} , and the Der eps 10^{-6} . Intensity computations were done with a Grid Size of 30 and the Gauss Kronrod integration algorithm.

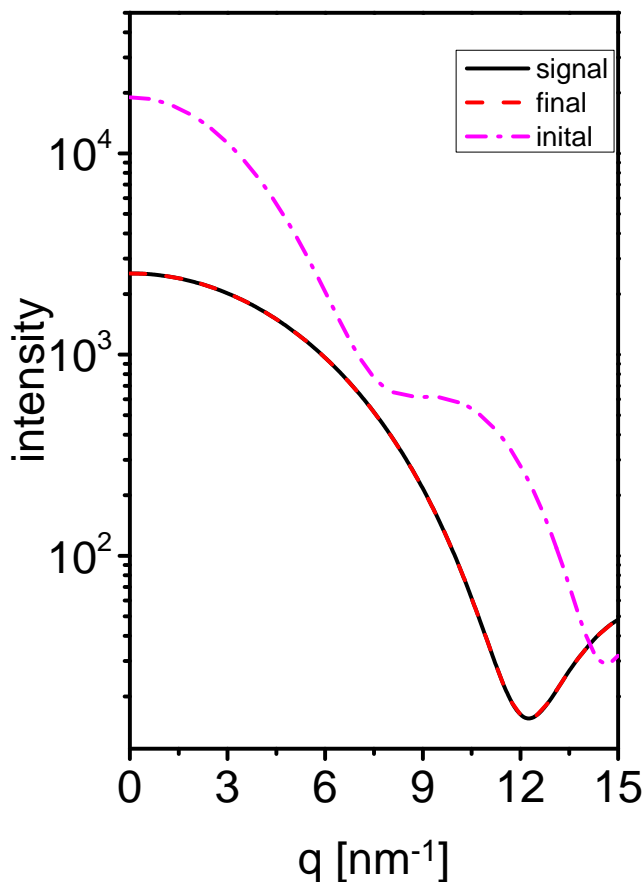


Fig. S35. Fitting to a modeled X-ray scattering intensity curve from a model of a uniform rod surrounded by two concentric uniform cylindrical shells. The structures were assumed to adopt isotropic distributed in random orientations in solution. The radius of the internal rod is $(0.15)0.1$ nm and its electron density is $400 \text{ e}/\text{nm}^3$. The first cylindrical shell is $(0.2)0.1$ nm thick, and its electron density is $600 \text{ e}/\text{nm}^3$. The second cylindrical shell is $(0.15)0.1$ nm thick, and its electron density is $800 \text{ e}/\text{nm}^3$. The solid black curve was computed in D+, and used as the target signal for the fitting. The broken magenta curve was computed by D+, using the initial values in parenthesis and the red broken curve is the result of the fit. Fitting was done using Levenberg-Marquardt algorithm, Ratio Residuals, Trivial Loss, with 20 iterations. Step Size was 10^{-5} , the Convergence tolerance was 10^{-6} , and Der eps was 10^{-6} . Intensity computations were done with a Grid Size of 30 and the Gauss Kronrod integration algorithm.

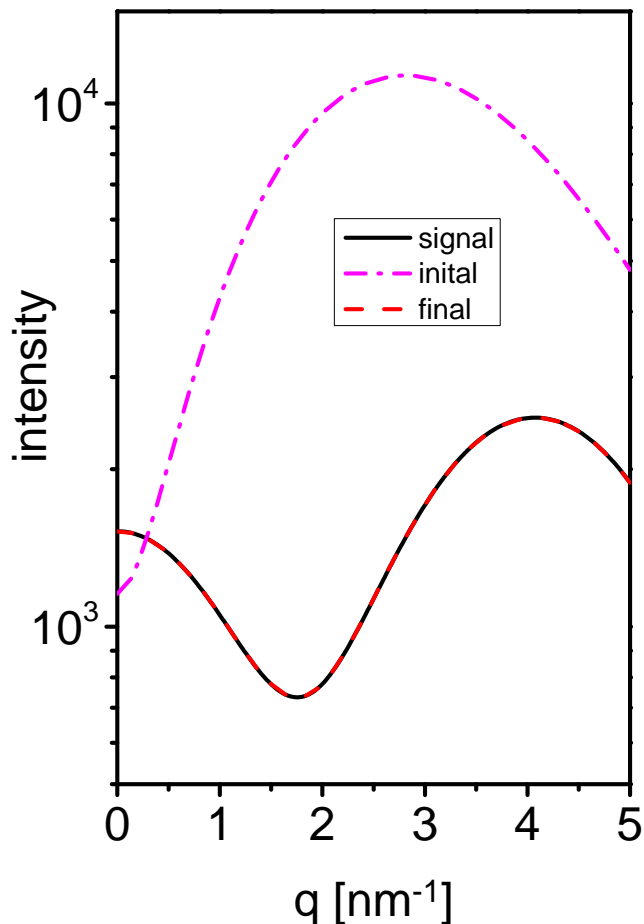


Fig. S36.

Fitting to a modeled X-ray scattering intensity curve from asymmetric slab layers. The structures were assumed to adopt isotropic distributed in random orientations in solution. The slabs were $1 \times 1 \text{ nm}^2$ in x-y plane and in the z direction the inner slab layer was 1 nm thick and electron density of $(250)280 \text{ e/nm}^3$. The layer was sandwiched between two outer 0.5 nm thick slabs, one with an electron density of $(300)400 \text{ e/nm}^3$ and the other with an electron density of $(600)450 \text{ e/nm}^3$. The solid black curve was computed in D+, and used as the target signal for the fitting. The broken magenta curve was computed by D+, using the initial values in parenthesis and the broken red curve is the result of the fit. Fitting was done using Levenberg-Marquardt method, Ratio Residuals, Trivial Loss, with 50 iterations. The Step Size was 10^{-5} , the Convergence tolerance was 10^{-6} , and Der eps 10^{-6} . Intensity computations were done with a Grid Size of 50 and the Gauss Kronrod integration algorithm.

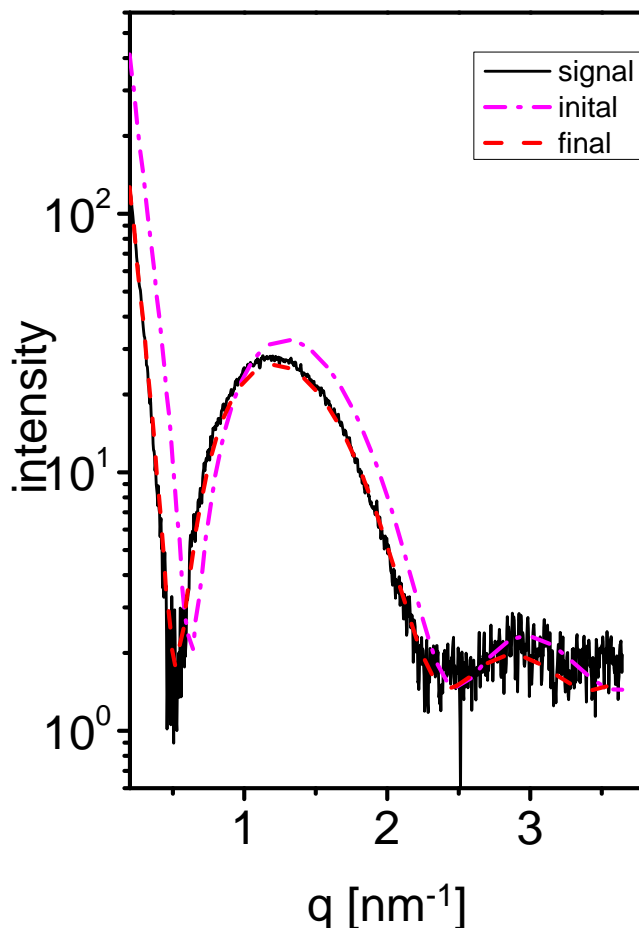


Fig. S37. Fitting of a modeled X-ray scattering intensity curve from symmetric slab layers to experimental scattering data from a dispersion of 30 mg/ml 1,2-dipalmitoyl-sn-glycero-3-phosphocholine (DPPC) lipid bilayers. The modeled structures were assumed to adopt isotropic distributed in random orientations in solution. In the model, the size of the slabs was 100 nm \times 100 nm in the x-y plane and in the z direction a (0.6)0.66 nm thick inner layer with an electron density of 139 e/nm³ was sandwiched between two outer slabs. Each outer slab had two layers, the first, was (1.3)1.44 nm thick and had an electron density of 335 e/nm³, and the second was (0.9)0.66 nm thick and had an electron density of 540 e/nm³. The solid black curve was measured in our in-house rotating anode setup (Nadler *et al.*, 2011), and used as the signal for the fitting. The broken magenta curve was computed by D+, using the initial values in parenthesis and the broken red curve is the result of the fit. Fitting was done using Levenberg-Marquardt algorithm, Ratio Residuals, Trivial Loss, and 5 iterations. Step Size was 0.1, the Convergence tolerance was 10⁻⁶, and Der eps 10⁻⁶. Intensity computations were done with a Grid Size of 200 and the adaptive (VEGAS) Monte Carlo integration algorithm.

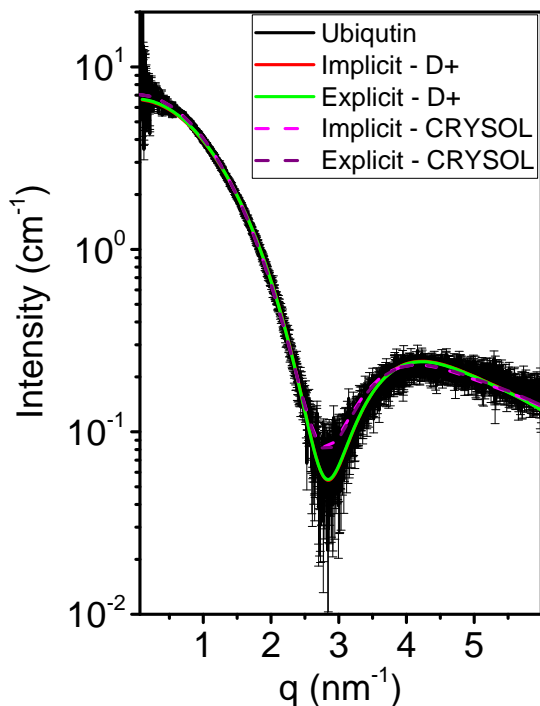


Fig. S38. Solution X-ray scattering data and models of Ubiquitin. Data taken from SASDAQ2 in the SASBDB database (Valentini *et al.*, 2014) are shown in black. The computed scattering curves are based on PDB ID 1UBQ. Hydrogen atoms were either implicitly taken into account or explicitly added to the PDB file by PyMol (Schrödinger, 2015). The broken scattering curves were computed by CRY SOL (ATSAS 2.8.2) (Svergun *et al.*, 1995), where the maximum order of harmonics was 50 and the order of Fibonacci grid was 18. The solid curves were computed by D+. In both D+ and CRY SOL the bulk water electron density was set to $334 \text{ e} \cdot \text{nm}^{-3}$ and the contribution of the solvation layer was taken into account. In D+, the **Grid Size** was 80, orientation average was computed by Monte Carlo integration, using Mersenne Twister algorithm. The solvent **Voxel Size** was 0.05 nm, the solvent **Probe Radius** was 0.14 nm and **Dummy Atoms** were used to account for the solvent excluded volume (see §5.5 in the main paper and Eq. 63). The fitting algorithm used the **Normal Residuals** cost function and the **Trivial Loss** function, with 20 iterations, **Step Size** of 0.01, **Convergence** tolerance of 0.01, and **Der eps** 0.1 (see §9.1 in the main paper). In the implicit model, the solvation thickness was (0.3)0.294 nm and its electron density was (364)336.6 $\text{e} \cdot \text{nm}^{-3}$. The excluded volume parameter, c_1 , in Eq. 60 was (1)1.014, and $R^2 = 0.941$. In the explicit model, c_1 was (1)1.012, the solvation thickness was (0.3)0.291 nm, its electron density was (364)335 $\text{e} \cdot \text{nm}^{-3}$, and $R^2 = 0.943$. In CRY SOL, in both models, the solvation thickness was 0.3 nm and the electron density of the solvation shell was $347 \text{ e} \cdot \text{nm}^{-3}$. In the implicit model $\chi^2 = 8.1$ and in the explicit model $\chi^2 = 8.0$.

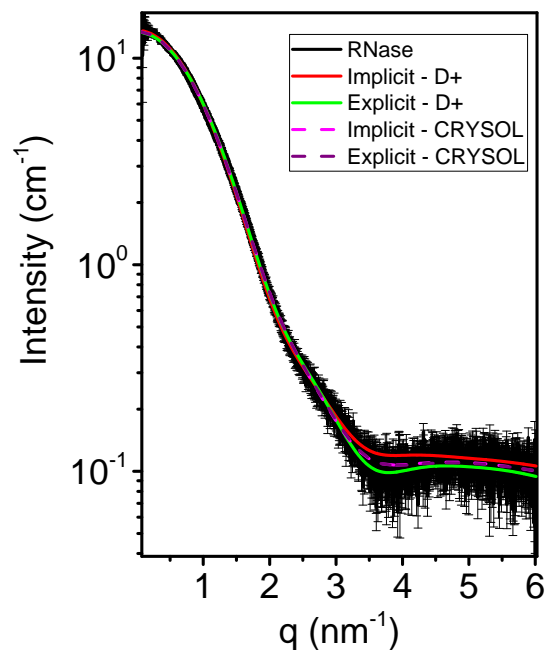


Fig. S39. Solution X-ray scattering data and models of RNase. Data taken from SASDAR2 in the SASBDB database (Valentini *et al.*, 2014) are shown in black. The computed scattering curves are based on PDB ID 1C0B. Hydrogen atoms were either implicitly taken into account or explicitly added to the PDB file by PyMol (Schrödinger, 2015). The broken scattering curves were computed by CRY SOL (ATSAS 2.8.2) (Svergun *et al.*, 1995), where the maximum order of harmonics was 50 and the order of Fibonacci grid was 18. The solid curves were computed by D+. In both D+ and CRY SOL the bulk water electron density was set to $334 \text{ e} \cdot \text{nm}^{-3}$ and the contribution of the solvation layer was taken into account. In D+, the **Grid Size** was 80, orientation average was computed by Monte Carlo integration, using Mersenne Twister algorithm. The solvent **Voxel Size** was 0.05 nm, the solvent **Probe Radius** was 0.14 nm and **Dummy Atoms** were used to account for the solvent excluded volume (see §5.5 in the main paper and Eq. 63). The fitting algorithm used the **Normal Residuals** cost function and the **Trivial Loss** function, with 20 iterations, **Step Size** of 0.01, **Convergence** tolerance of 0.01, and **Der eps** 0.1 (see §9.1 in the main paper). In the implicit model, the solvation thickness was (0.3)0.292 nm and its electron density was (364)363.8 $\text{e} \cdot \text{nm}^{-3}$. The excluded volume parameter, c_1 , in Eq. 60 was (1)1.003, and $R^2 = 0.998$. In the explicit model, c_1 was (1)1.012, the solvation thickness was (0.3)0.305 nm, its electron density was (364)352 $\text{e} \cdot \text{nm}^{-3}$, and $R^2 = 0.997$. In CRY SOL, in both models, the solvation thickness was 0.3 nm. In the implicit model the electron density of the solvation shell was 374 $\text{e} \cdot \text{nm}^{-3}$, and $\chi^2 = 2.67$. In the explicit model and the electron density of the solvation shell was 371 $\text{e} \cdot \text{nm}^{-3}$, and $\chi^2 = 2.65$.

10. Solvation layers of complexes

Solvation layers of large complexes can be computed in $D+$, in a scalable manner, based on the solvation layers of each of the subunits. Corrections, however, have to be included if the solvation layers of the subunits overlap when docked into the complex structure (Fig. S40). The errors can be corrected by subtracting the overlapping solvation parts.

We can demonstrate the approach with a simple case where three identical subunits form a chain, with overlaps in their solvation layer (Fig. S40). The solvation shell overlap of the central subunit is different than that of the subunits at the edges.

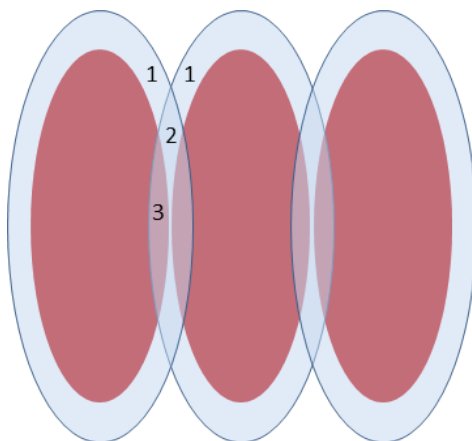


Fig. S40. A short chain containing three adjacent subunits. Each subunit is surrounded by a solvation layer, assuming it was isolated. The numbers correspond to different solvation amplitude scales. If the scale of the solvation layer is set to 1, then area 1 corresponds to a scale of 1, area 2 corresponds to a scale of 2, and area 3 corresponds to a scale of 1. Area 3 also contains the contribution from the subunit amplitude.

In $D+$, we can compute the amplitude of a monomer subunit, surrounded by a solvation layer ($f_{\text{Solvated Monomer}}$). The Solvated subunit can then be placed in a symmetry containing two repeats next to one another (Fig. S41). The scattering amplitude is then $f_{\text{Solvated Monomer}} \left(e^{i\vec{q}\cdot\vec{r}_1} + e^{i\vec{q}\cdot\vec{r}_2} \right)$. We can also compute the solvation layer amplitude, which is surrounding a single structure, containing the same two subunits next to one another, $f_{\text{Solvated Dimer}}$ (Fig. S42). In the latter case there are no overlapping

solvation layers.

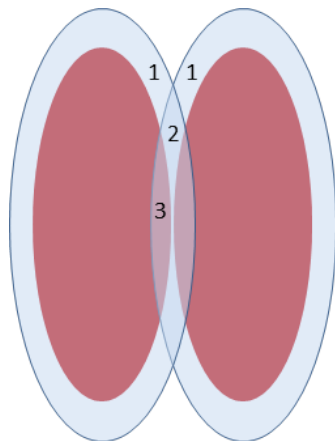


Fig. S41. Two adjacent subunits, where each subunit is surrounded by its own solvation layer (Eqs. S3 and S4). The numbers correspond to different types of overlapping regions, as in Fig. S40.

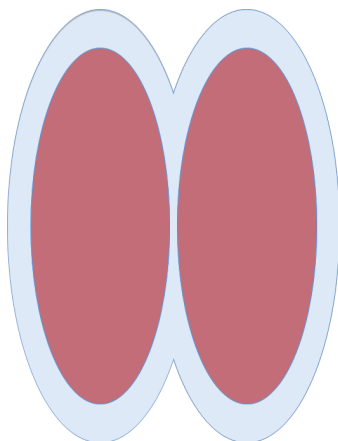


Fig. S42. Solvated dimer. The solvation layer surrounds a single structure made of two identical adjacent subunits. The scattering amplitude is $f_{\text{Solvated Dimer}}$.

To construct the interface between two monomers, three amplitudes should be calculated: one for each solvated monomer $f_{\text{Solvated Monomer}}$, at its own position (Fig. S41 and Eqs. S3 and S4), and one for the solvated dimer, $f_{\text{Solvated Dimer}}$ (Fig. S42). The two monomers have the same magnitude of scattering amplitude but different phases.

If the position of the first monomer is at the origin, $\vec{r}_1 = 0$, and the second monomer is at $\vec{r}_2 = \vec{R}$, we get:

$$f_{\text{Solvated Monomer 1}} = f_{\text{Solvated Monomer}} \cdot e^{i\vec{q}\cdot\vec{r}_1} = f_{\text{Solvated Monomer}} \quad (\text{S3})$$

$$f_{\text{Solvated Monomer 2}} = f_{\text{Solvated Monomer}} \cdot e^{i\vec{q}\cdot\vec{r}_2} = f_{\text{Solvated Monomer}} \cdot e^{i\vec{q}\cdot\vec{R}} \quad (\text{S4})$$

The solvation correction amplitude is then given by:

$$F_{\text{Solvation correction}} = f_{\text{Solvated Dimer}} - (f_{\text{Solvated Monomer 1}} + f_{\text{Solvated Monomer 2}}) \quad (\text{S5})$$

Fig. S43 shows the result of Eq. S5, when the scale of each amplitude is set to 1.

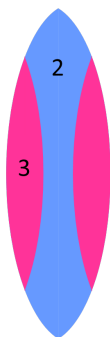


Fig. S43. The correction amplitude obtained from Eq. S5. If the scale of the solvation layer is set to 1, then the scale of areas 2 and 3 is -1.

The solvated trimer is computed from three solvated monomers at $\vec{r}_1 = 0$, $\vec{r}_2 = \vec{R}$, and $\vec{r}_3 = 2\vec{R}$, and two Solvation corrections:

$$f_{\text{Solvated Trimer}} = f_{\text{Solvated Monomer}} \left(e^{i\vec{q}\cdot\vec{r}_1} + e^{i\vec{q}\cdot\vec{r}_2} + e^{i\vec{q}\cdot\vec{r}_3} \right) + F_{\text{Solvation correction}} \left(e^{i\vec{q}\cdot\vec{r}_1} + e^{i\vec{q}\cdot\vec{r}_2} \right) \quad (\text{S6})$$

$$= f_{\text{Solvated Monomer}} \left(1 + e^{i\vec{q}\cdot\vec{R}} + e^{i2\vec{q}\cdot\vec{R}} \right) + F_{\text{Solvation correction}} \left(1 + e^{i\vec{q}\cdot\vec{R}} \right) \quad (\text{S7})$$

Fig. S44 shows the resulting solvated trimer computed in Eq. S7.

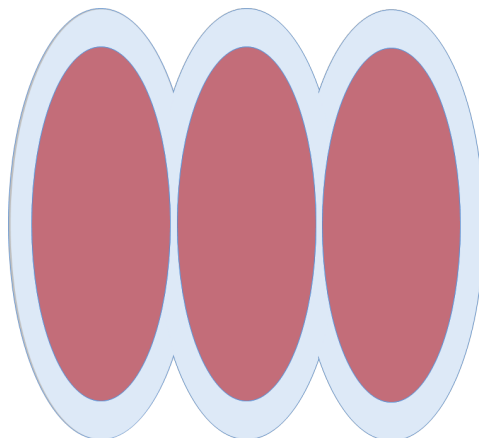


Fig. S44. The solvated trimer computed by Eq. S7.

To apply the correction in a symmetry, the repeating subunit should already include the corrections at each one of the interfaces with its neighboring subunits. This principle is demonstrated in Fig. S45 and in Eq. S8:

$$f_{\text{Corrected Subunit}} = f_{\text{Solvated Monomer}} + \frac{1}{2} \times F_{\text{Centered Solvation correction}} \left(e^{\frac{-i\vec{q}\cdot\vec{R}}{2}} + e^{\frac{i\vec{q}\cdot\vec{R}}{2}} \right) \quad (\text{S8})$$

where $F_{\text{Centered Solvation correction}}$ is $F_{\text{Solvation correction}}$, which was placed at the origin:

$$F_{\text{Centered Solvation correction}} = F_{\text{Solvation correction}} \cdot e^{\frac{-i\vec{q}\cdot\vec{R}}{2}}$$

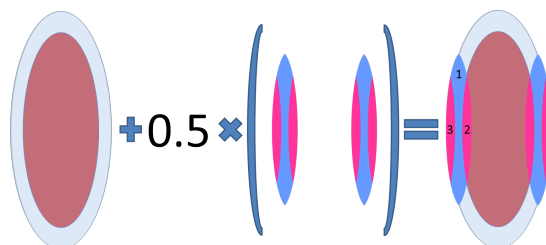


Fig. S45. Modified subunit to be used in a symmetry. The solvation correction amplitude is added to the repeating subunit with a factor of 0.5, at each end of the monomer, using Eq. S8). If the scale of the solvation layer is set to 1, then the amplitude scale of area 1 and area 3 is 0.5, whereas the scale of area 2 is -0.5.

Note that within a given assembly symmetry, the interfaces may vary in different subunit locations. In this case, the symmetry may be decomposed into smaller symmetries where each has the original subunit with a different solvation correction.

Going back to our example, the modified subunit is then inserted into its assembly symmetry. In the example in Fig. S46 we get:

$$f_{\text{Corrected Subunit in Symmetry}} = f_{\text{Corrected Subunit}} \left(1 + e^{i\vec{q}\cdot\vec{R}} + e^{i2\vec{q}\cdot\vec{R}} \right) \quad (\text{S9})$$

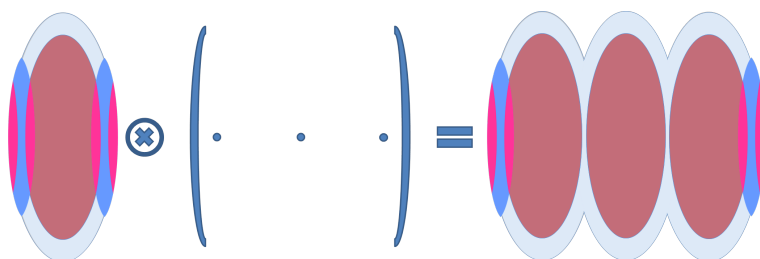


Fig. S46. Modified subunit in a symmetry. The cross product corresponds to convolution of the subunit with a sum of three delta functions, whose centers are where the dots are located. Inserting the corrected subunit from Fig. S45 to the trimer symmetry results in a trimer, with corrections at each one of the chain edges.

The edges, whose interfaces are uncoupled to neighboring subunits at one of the two ends, should be further corrected:

$$f_{\text{Solvated Dimer}} = f_{\text{Corrected Subunit in Symmetry}} - 0.5 \cdot F_{\text{Centered Solvation correction}} \left(e^{\frac{-i\vec{q}\cdot\vec{R}}{2}} + e^{\frac{i5\vec{q}\cdot\vec{R}}{2}} \right) \quad (\text{S10})$$

Fig. S47 demonstrates this final correction.

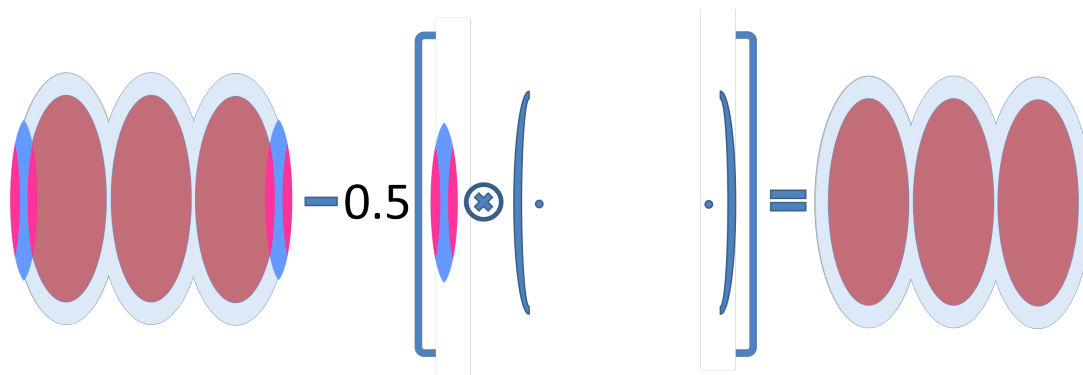


Fig. S47. Edge corrections. The structure in Fig. S46) is corrected to account for subunits located at the edge of the symmetry.

10.1. Solvated tubulin ring

We applied the approach described in the earlier section, to compute the scattering amplitude of the solvation layer of a tubulin ring, using the amplitude of a solvated tubulin dimer in horizontal orientation and the relevant correction terms. According to Eq. S5 a solvation correction amplitude was computed and added to the amplitude of the solvated dimer (Fig. S48):

$$f_{\text{Corrected Solvated Horizontal Dimer}} = f_{\text{Solvated Horizontal Dimer}} + F_{\text{solvation Horiznotal correction}} \quad (\text{S11})$$



Fig. S48. Solvation correction for a tubulin dimer in horizontal orientation (as in a tubulin ring).

To obtain the scattering amplitude of the solvated tubulin ring, the corrected solvated tubulin dimer amplitude was inserted into the ring symmetry (Fig. S49):

$$f_{\text{Corrected Solvation Ring}} = \sum_{i=0}^{12} f_{\text{Corrected Solvated Horizontal Dimer}} \left(\mathbf{A}_{\mathbf{n}}^{-1} \vec{q} \right) e^{i\vec{q} \cdot \vec{R}_n} \quad (\text{S12})$$

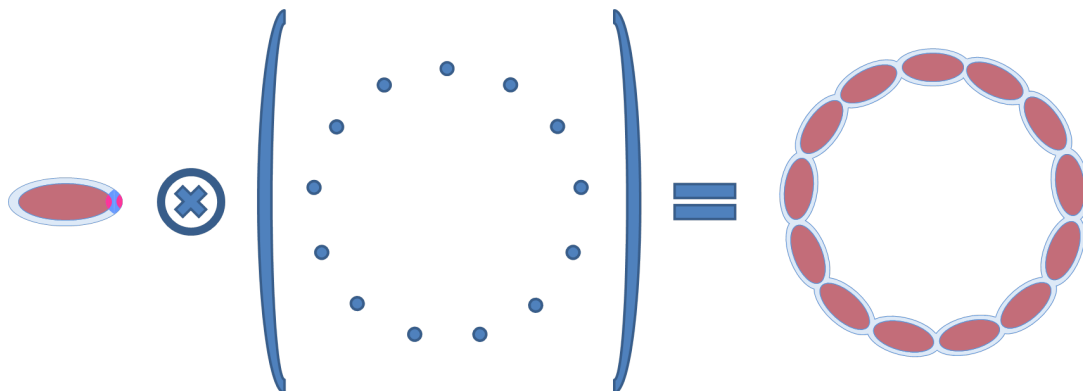


Fig. S49. Solvated tubulin dimer after solvation correction placed in the symmetry of tubulin ring.

The ring rotation matrices, A_n , are defined as:

$$\mathbf{A}_{\mathbf{n}} = \begin{pmatrix} \cos\left(\frac{360}{13}n\right) & -\sin\left(\frac{360}{13}n\right) & 0 \\ \sin\left(\frac{360}{13}n\right) & \cos\left(\frac{360}{13}n\right) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{S13})$$

$$\mathbf{A}_{\mathbf{n}}^{-1} = \begin{pmatrix} \frac{\cos\left(\frac{360}{13}n\right)}{\cos^2\left(\frac{360}{13}n\right) + \sin^2\left(\frac{360}{13}n\right)} & \frac{\sin\left(\frac{360}{13}n\right)}{\cos^2\left(\frac{360}{13}n\right) + \sin^2\left(\frac{360}{13}n\right)} & 0 \\ -\frac{\sin\left(\frac{360}{13}n\right)}{\cos^2\left(\frac{360}{13}n\right) + \sin^2\left(\frac{360}{13}n\right)} & \frac{\cos\left(\frac{360}{13}n\right)}{\cos^2\left(\frac{360}{13}n\right) + \sin^2\left(\frac{360}{13}n\right)} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{S14})$$

The location vectors, R_n , are defined as:

$$\vec{R}_n = \begin{pmatrix} R \sin\left(\frac{360}{13}n\right) \\ R \cos\left(\frac{360}{13}n\right) \\ 0 \end{pmatrix} \quad (\text{S15})$$

where $n \in [0, 1, \dots, 12]$. As the ring structure is circular, no edge correction are required.

To test our approach, we compared the results of Eq. S12 with the intensity of solvated tubulin ring that was computed from a single PDB file, containing the atoms of the entire ring.

Fig. S50 shows that both methods give similar scattering intensity curves for solvated tubulin ring.

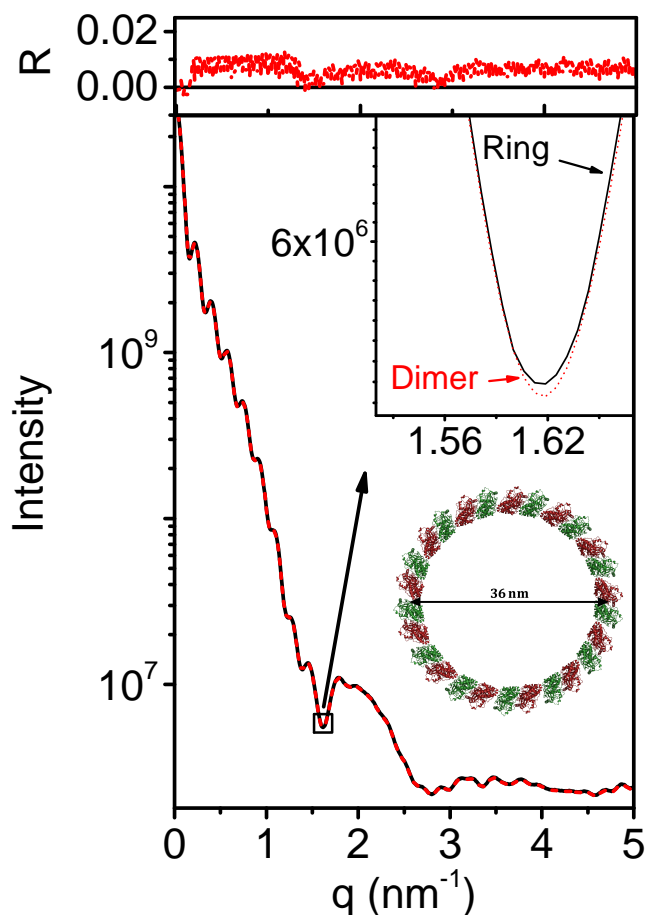


Fig. S50. Computed X-ray scattering intensity from an atomic model of solvated tubulin ring, shown as a ribbon diagram. The rings were assumed to adopt isotropic distributed in random orientations in solution. H atoms were explicitly added to the PDB file (Chen *et al.*, 2010). The solid black curve was computed by D+, using the PDB file of the entire ring. The broken red curve was computed by D+ from the PDB file of the dimer (PDB ID 3J6F) with its solvation corrections and docking the result into the symmetry of the ring (see text). The solvation layers were 0.3 nm thick and their electron density contrasts (with respect to bulk water) were $30 e/\text{nm}^3$. Both values are often used for computing the scattering amplitude contribution from solvation layers of proteins (Svergun *et al.*, 1995). The inset shows a small part of the curve on an expanded scale. The top curves show the normalized residual, R , where the reference curve is the intensity computed from the PDB file of the entire ring (Eq. S2).

Fig. S51 compares the contribution of the solvation layer using the two methods and further confirms that both computations yield very similar results.

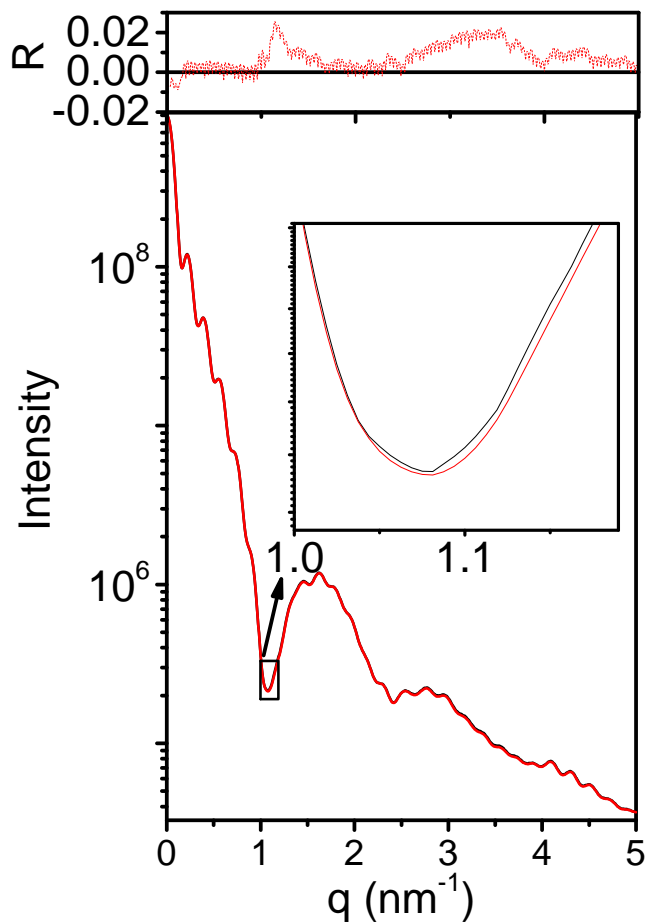


Fig. S51. The contribution of the solvation layers to the scattering curves in Fig. S50. Black curve is based on the PDB file of the entire ring. Broken red curve is based on the modified tubulin dimer (PDB ID 3J6F). The solvation layers were 0.3 nm thick and their electron density contrasts (with respect to bulk water) were $30 \text{ e}/\text{nm}^3$. The inset shows a small part of the curve on an expanded scale. The top curves show the normalized residual, R , where the reference curve is the intensity computed by PDB file of the entire ring (Eq. S2).

10.2. Solvated microtubule

The approach that was used for computing the solvation layer of tubulin rings was adopted and adjusted for computing the scattering from solvated MT (Fig. 6). A few

solvation overlap corrections were created and docked at the relevant locations with the MT lattice. The MT lattice symmetry is described using the following rotation matrices and translation vectors:

$$\mathbf{A}_n^{\text{MT}} = \begin{pmatrix} \cos\left(\frac{360}{N_{\text{PF Number}}}n\right) & -\sin\left(\frac{360}{N_{\text{PF Number}}}n\right) & 0 \\ \sin\left(\frac{360}{N_{\text{PF Number}}}n\right) & \cos\left(\frac{360}{N_{\text{PF Number}}}n\right) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{S16})$$

$$\vec{R}_{j,n}^{\text{MT}} = \begin{pmatrix} R \sin\left(\frac{360}{N_{\text{PF Number}}}n\right) \\ R \cos\left(\frac{360}{N_{\text{PF Number}}}n\right) \\ H_{j-1} + P_n \end{pmatrix} \quad (\text{S17})$$

where $n \in [1, \dots, N_{\text{PF Number}}]$ and $H_j = K \cdot P \cdot j$ for $j \in [1, \dots, N_{\text{Long}}]$. N_{Long} is the number of longitudinal repeats of the subunit in the total helix. K is one over the number of helix starts, which is 3 for the monomers (assuming there is no difference between the α and β monomers) and $1\frac{1}{2}$ for the dimers. $P_n = \frac{P}{N_{\text{PF Number}}} \cdot n$, where $N_{\text{PF Number}}$ is the number of protofilaments in the MT (or the number of dimers per pitch), P is the length of the pitch, defined as the axial rise along the \hat{z} direction per $N_{\text{PF Number}}$ dimers. $P = 0$ corresponds to a ring.

Unlike tubulin ring, MT is not a fully closed structure and it has a seam. Therefore, different corrections were added to dimers that were located at the center of the MT lattice, or at one of its edges, or along the MT seam (see marked dimers in Fig. S52).

The center of mass of the tubulin dimer was first shifted to the origin by computing

$$\rho_0(\vec{r}) = \rho(\vec{r} - \vec{R}_{\text{c.m.}}),$$

where $\rho(\vec{r})$ represents the electron density of the dimer (given by PDB ID 3J6F), and $\vec{R}_{\text{c.m.}}$ is its center of mass. Solvation correction for the dimers that are located side by side within the central part of the MT lattice, was constructed by placing the dimer in the MT symmetry for subunits whose indexes are $j = 1$ and $n \in [1, 2]$, giving

$$\rho_{1,1}^{\text{Dimer Side}}(\vec{r}) = \rho_0 \left[\left(\mathbf{A}_1^{\text{MT}} \right)^{-1} \left(\vec{r} - \vec{R}_{1,1}^{\text{MT}} \right) \right] \text{ and } \rho_{1,2}^{\text{Dimer Side}}(\vec{r}) = \rho_0 \left[\left(\mathbf{A}_2^{\text{MT}} \right)^{-1} \left(\vec{r} - \vec{R}_{1,2}^{\text{MT}} \right) \right],$$

respectively. The electron density of the pair of dimers, $\rho_{1,1 \& 1,2}^{\text{Dimer Pair Side}}(\vec{r}) = \rho_{1,1}^{\text{Dimer Side}}(\vec{r}) + \rho_{1,2}^{\text{Dimer Side}}(\vec{r})$, was then aligned back to $\rho_0(\vec{r})$ (the initial dimer):

$$\begin{aligned} \rho_{1,1 \& 1,2}^{\text{Aligned Dimer Pair Side}}(\vec{r}) &= \rho_{1,1 \& 1,2}^{\text{Dimer Pair Side}} \left[\mathbf{A}_1^{\text{MT}} \vec{r} + \vec{R}_{1,1}^{\text{MT}} \right] = \\ &= \rho_{1,1}^{\text{Dimer Side}} \left[\mathbf{A}_1^{\text{MT}} \vec{r} + \vec{R}_{1,1}^{\text{MT}} \right] + \rho_{1,2}^{\text{Dimer Side}} \left[\mathbf{A}_1^{\text{MT}} \vec{r} + \vec{R}_{1,1}^{\text{MT}} \right] \\ &\equiv \rho_{1,1}^{\text{Aligned Dimer Side}}(\vec{r}) + \rho_{1,2}^{\text{Aligned Dimer Side}}(\vec{r}) \end{aligned}$$

so the electron density (given as a PDB file) of the aligned $\rho_{1,1}^{\text{Dimer Side}}(\vec{r})$, which we shall refer to as $\rho_{1,1}^{\text{Aligned Dimer Side}}(\vec{r})$, overlapped with $\rho_0(\vec{r})$. Therefore the side (or lateral) solvation correction is:

$$\begin{aligned} f_{\text{Side Solvation Correction}} &= f_{1,1 \& 1,2}^{\text{Solvated Aligned Dimer Pair Side}} - \\ &= \left(f_{1,1}^{\text{Solvated Aligned Dimer Side}} + f_{1,2}^{\text{Solvated Aligned Dimer Side}} \right) \end{aligned}$$

where f is the scattering amplitude of the corresponding solvated structures defined above.

To create the correction for dimers that are located on top of one another within the central part of the MT lattice, the same process was repeated for two dimers with indexes $j \in [1, 2]$ and $n = 1$, giving $\rho_{1,1}^{\text{Dimer Top}}(\vec{r}) = \rho_0 \left[\left(\mathbf{A}_1^{\text{MT}} \right)^{-1} \left(\vec{r} - \vec{R}_{1,1}^{\text{MT}} \right) \right]$ and $\rho_{2,1}^{\text{Dimer Top}}(\vec{r}) = \rho_0 \left[\left(\mathbf{A}_1^{\text{MT}} \right)^{-1} \left(\vec{r} - \vec{R}_{2,1}^{\text{MT}} \right) \right]$, respectively. The electron density (given as a PDB file) of the pair of dimers, $\rho_{1,1 \& 2,1}^{\text{Dimer Pair Top}}(\vec{r}) = \rho_{1,1}^{\text{Dimer Top}}(\vec{r}) + \rho_{2,1}^{\text{Dimer Top}}(\vec{r})$, was then aligned back to $\rho_0(\vec{r})$, the initial dimer: $\rho_{1,1 \& 2,1}^{\text{Aligned Dimer Pair Top}}(\vec{r}) = \rho_{1,1 \& 2,1}^{\text{Dimer Pair Top}} \left[\mathbf{A}_1^{\text{MT}} \vec{r} + \vec{R}_{1,1}^{\text{MT}} \right]$ so the electron density (given as a PDB file) of the aligned $\rho_{1,1}^{\text{Dimer Top}}(\vec{r})$, which we shall refer to as $\rho_{1,1}^{\text{Aligned Dimer Top}}(\vec{r})$, overlapped with $\rho_0(\vec{r})$. Similarly, $\rho_{2,1}^{\text{Aligned Dimer Top}}(\vec{r}) = \rho_{2,1}^{\text{Dimer Top}} \left[\mathbf{A}_1^{\text{MT}} \vec{r} + \vec{R}_{1,1}^{\text{MT}} \right]$. Therefore the top (or vertical) solvation correction is:

$$\begin{aligned} f_{\text{Top Solvation Correction}} &= f_{1,1 \& 2,1}^{\text{Solvated Aligned Dimer Pair Top}} - \\ &= \left(f_{1,1}^{\text{Solvated Aligned Dimer Top}} + f_{2,1}^{\text{Solvated Aligned Dimer Top}} \right) \end{aligned}$$

where f is the scattering amplitude of the relevant solvated structures.

Two additional corrections were required for the MT seam, in which α - and β -tubulin are laterally located next to one another (instead of having α - next to α -tubulin and β - next to β -tubulin, in the central MT lattice). The above correction process was repeated for two monomers of the seam. Three dimers with indexes ($j = 1, n = N_{\text{PF Number}}$), ($j = 2, n = 1$) and ($j = 3, n = 1$), giving

$$\rho_{1, N_{\text{PF Number}}}^{\text{Dimer Seam}}(\vec{r}) = \rho_0 \left[\left(\mathbf{A}_{N_{\text{PF Number}}}^{\text{MT}} \right)^{-1} \left(\vec{r} - \vec{R}_{1, N_{\text{PF Number}}}^{\text{MT}} \right) \right],$$

$$\rho_{2,1}^{\text{Dimer Seam}}(\vec{r}) = \rho_0 \left[\left(\mathbf{A}_1^{\text{MT}} \right)^{-1} \left(\vec{r} - \vec{R}_{2,1}^{\text{MT}} \right) \right]$$

and $\rho_{3,1}^{\text{Dimer Seam}}(\vec{r}) = \rho_0 \left[\left(\mathbf{A}_1^{\text{MT}} \right)^{-1} \left(\vec{r} - \vec{R}_{3,1}^{\text{MT}} \right) \right]$, respectively. From dimers $\rho_{2,1}^{\text{Dimer Seam}}(\vec{r})$ and $\rho_{3,1}^{\text{Dimer Seam}}(\vec{r})$, two monomers were joined to form a new dimer (in reverse order) - β -tubulin of dimer ($j = 2, n = 1$), $\rho_{2,1,\beta}^{\text{Dimer Seam}}(\vec{r})$ and α -tubulin of dimer ($j = 3, n = 1$), $\rho_{3,1,\alpha}^{\text{Dimer Seam}}(\vec{r})$.

The newly formed dimer, $\rho_{2,1,\beta \& 3,1,\alpha}^{\text{Dimer Seam}}(\vec{r})$ was combined with ($j = 1, n = N_{\text{PF Number}}$) to form the PDB file that describes the electron density of the pair of dimers,

$$\rho_{1, N_{\text{PF Number}} \& 2,1,\beta \& 3,1,\alpha}^{\text{Dimer Pair Seam}}(\vec{r}) = \rho_{1, N_{\text{PF Number}}}^{\text{Dimer Seam}}(\vec{r}) + \rho_{2,1,\beta \& 3,1,\alpha}^{\text{Dimer Seam}}(\vec{r}).$$

The pair was then aligned back to $\rho_0(\vec{r})$ (the initial dimer):

$$\rho_{1, N_{\text{PF Number}} \& 2,1,\beta \& 3,1,\alpha}^{\text{Aligned Dimer Pair Seam}}(\vec{r}) = \rho_{1, N_{\text{PF Number}} \& 2,1,\beta \& 3,1,\alpha}^{\text{Dimer Pair Seam}} \left[\mathbf{A}_{N_{\text{PF Number}}}^{\text{MT}} \vec{r} + \vec{R}_{1, N_{\text{PF Number}}}^{\text{MT}} \right]$$

so the electron density (given as a PDB file) of the aligned $\rho_{1, N_{\text{PF Number}}}^{\text{Dimer Seam}}(\vec{r})$, which we shall refer to as $\rho_{1, N_{\text{PF Number}}}^{\text{Aligned Dimer Seam}}(\vec{r})$, overlapped with $\rho_0(\vec{r})$. Similarly,

$$\rho_{2,1,\beta \& 3,1,\alpha}^{\text{Aligned Dimer Seam}}(\vec{r}) = \rho_{2,1,\beta \& 3,1,\alpha}^{\text{Dimer Seam}} \left[\mathbf{A}_{N_{\text{PF Number}}}^{\text{MT}} \vec{r} + \vec{R}_{1, N_{\text{PF Number}}}^{\text{MT}} \right].$$

From the electron density (given as a PDB file) of $\rho_{1, N_{\text{PF Number}} \& 2,1,\beta \& 3,1,\alpha}^{\text{Aligned Dimer Pair Seam}}(\vec{r})$, two electron densities (given by two PDB files) were extracted - the bottom α and β

monomers, $\rho_{1, N_{\text{PF Number}}, \alpha \& 2,1,\beta}^{\text{Aligned Bottom Monomer Pair Seam}}(\vec{r})$ and the top α and β monomers, $\rho_{1, N_{\text{PF Number}}, \beta \& 3,1,\alpha}^{\text{Aligned Top Monomer Pair Seam}}(\vec{r})$.

Each one of the new PDB files was further separated into their corresponding α - and the β - tubulin monomers, resulting in four PDB files describing the following electron densities: $\rho_{1,N_{\text{PF Number}},\alpha}^{\text{Aligned Bottom Monomer Seam}}(\vec{r})$, $\rho_{2,1,\beta}^{\text{Aligned Bottom Monomer Seam}}(\vec{r})$, $\rho_{1,N_{\text{PF Number}},\beta}^{\text{Aligned Top Monomer Seam}}(\vec{r})$, and $\rho_{3,1,\alpha}^{\text{Aligned Top Monomer Seam}}(\vec{r})$. Therefore the monomer seam solvation corrections are:

$$f_{\text{Solvation Correction Bottom Seam Monomer}} = f_{1,N_{\text{PF Number}}\alpha\& 2,1\beta}^{\text{Solvated Aligned Bottom Monomer Pair Seam}} - \left(f_{1,N_{\text{PF Number}}\alpha}^{\text{Solvated Aligned Bottom Monomer Seam}} + f_{2,1\beta}^{\text{Solvated Aligned Bottom Monomer Seam}} \right)$$

$$f_{\text{Solvation Correction Top Seam Monomer}} = f_{1,N_{\text{PF Number}}\beta\& 3,1\alpha}^{\text{Solvated Aligned Top Monomer Pair Seam}} - \left(f_{1,N_{\text{PF Number}}\beta}^{\text{Solvated Aligned Top Monomer Seam}} + f_{3,1\alpha}^{\text{Solvated Aligned Top Monomer Seam}} \right)$$

where f is the scattering amplitude of the relevant solvated structures.

The entire solvated MT was computed by docking the amplitude of the solvated tubulin dimer, $f_{\text{Solvated Tubulin Dimer}}$, into the MT symmetry, with $n \in [1, \dots, N_{\text{PF Number}}]$ and $j \in [1, \dots, 48]$. Solvation correction amplitudes were then docked into the MT symmetry as follows. Side solvation correction amplitude, $f_{\text{Side Solvation Correction}}$, was docked into the MT symmetry with $n \in [1, \dots, N_{\text{PF Number}} - 1]$ and $j \in [1, \dots, 48]$. Top Solvation correction amplitude, $f_{\text{Top Solvation Correction}}$, was docked into the MT symmetry with $n \in [1, \dots, N_{\text{PF Number}}]$ and $j \in [1, \dots, 47]$. Bottom seam solvation correction amplitude, $f_{\text{Solvation Correction Bottom Seam Monomer}}$, was docked into the MT symmetry with $n = N_{\text{PF Number}}$ and $j \in [1, \dots, 47]$. Top seam solvation correction amplitude, $f_{\text{Solvation Correction Top Seam Monomer}}$, was docked into the MT symmetry with $n = N_{\text{PF Number}}$ and $j \in [1, \dots, 46]$.

The scattering amplitude of the solvated MT is then:

$$\begin{aligned}
 f_{\text{Solvated MT}} = & \sum_{n=1}^{N_{\text{PF Number}}} \sum_{j=1}^{48} f_{\text{Solvated Tubulin Dimer}} \left[\left(\mathbf{A}_{\mathbf{n}}^{\text{MT}} \right)^{-1} \vec{q} \right] e^{i\vec{q} \cdot \vec{R}_{j,n}^{\text{MT}}} + \\
 & \sum_{n=1}^{N_{\text{PF Number}}-1} \sum_{j=1}^{48} f_{\text{Side Solvation Correction}} \left[\left(\mathbf{A}_{\mathbf{n}}^{\text{MT}} \right)^{-1} \vec{q} \right] e^{i\vec{q} \cdot \vec{R}_{j,n}^{\text{MT}}} + \\
 & \sum_{n=1}^{N_{\text{PF Number}}} \sum_{j=1}^{47} f_{\text{Top Solvation Correction}} \left[\left(\mathbf{A}_{\mathbf{n}}^{\text{MT}} \right)^{-1} \vec{q} \right] e^{i\vec{q} \cdot \vec{R}_{j,n}^{\text{MT}}} + \\
 & \sum_{j=1}^{47} f_{\text{Solvation Correction Bottom Seam Monomer}} \left[\left(\mathbf{A}_{N_{\text{PF Number}}}^{\text{MT}} \right)^{-1} \vec{q} \right] e^{i\vec{q} \cdot \vec{R}_{N_{\text{PF Number}}:j}^{\text{MT}}} + \\
 & \sum_{j=1}^{46} f_{\text{Solvation Correction Top Seam Monomer}} \left[\left(\mathbf{A}_{N_{\text{PF Number}}}^{\text{MT}} \right)^{-1} \vec{q} \right] e^{i\vec{q} \cdot \vec{R}_{N_{\text{PF Number}}:j}^{\text{MT}}}
 \end{aligned}$$

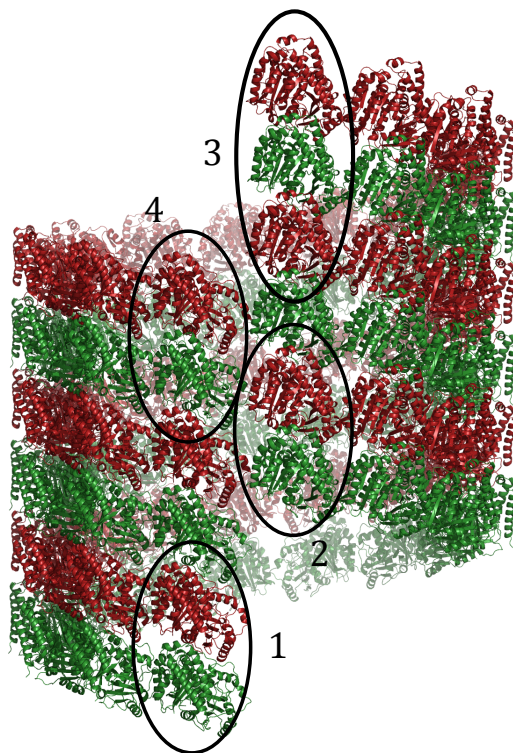


Fig. S52. Ribbon diagram of a short MT, where α -tubulin is in green and β -tubulin is in red. The numbers in the diagram refer to dimers with a different type of solvation correction. Four solvation corrections types were applied: side (as in Fig. S43), top (as in Fig. S48), bottom and top monomers of the MT seam. 4 dimers are marked: (1) The first dimer in the minus end of the MT lattice, which is corrected for top and side solvation overlap, as applied for most other dimers in the MT lattice. (2) A dimer, which is part of the seam, which is corrected for the top solvation overlap and in addition for the side of the top and bottom monomers solvation overlap. (3) Three final monomers with $n = N_{\text{PF Number}}$ in the plus end of the MT lattice, which do not need side solvation correction. The two top monomers also do not require top solvation overlap correction. (4) The top dimer with $n = 1$, to which only side solvation correction is applied.

References

- Alushin, G. M., Lander, G. C., Kellogg, E. H., Zhang, R., Baker, D. & Nogales, E. (2014). *Cell*, **157**(5), 1117–1129.
- Ben-Nun, T., Ginsburg, A., Székely, P. & Raviv, U. (2010). *Journal of Applied Crystallography*, **43**(6), 1522–1531.
- Chen, V. B., Arendall, III, W. B., Headd, J. J., Keedy, D. A., Immormino, R. M., Kapral, G. J., Murray, L. W., Richardson, J. S. & Richardson, D. C. (2010). *Acta Crystallographica Section D*, **66**(1), 12–21.
- Debye, P. (1915). *Ann. Phys.* **351**(6), 809–823.

- Diaz, F. J., Pantos, E., Bordas, J. & Andreu, J. M. (1994). *Journal of molecular biology*, **238**(2), 214–225.
- Ginsburg, A., Ben-Nun, T., Asor, R., Shemesh, A., Ringel, I. & Raviv, U. (2016). *Journal of Chemical Information and Modeling*, **56**(8), 1518–1527.
- Grudin, S., Garkavenko, M. & Kazennov, A. (2017). *Acta Crystallographica Section D: Structural Biology*, **73**(5).
- Hamilton, W. (1974). *Birmingham: Kynoch Press. (Present distributor Kluwer Academic Publishers, Dordrecht. Table 2.2B.)*, pp. 273–284.
- Heckbert, P. (1994). *Graphics Gems IV (IBM Version)*. Elsevier.
- Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library.
- Marsh, R. & Slagle, K. (1983). *Acta Crystallographica Section A: Foundations of Crystallography*, **39**(1), 173–173.
- Nadler, M., Steiner, A., Dvir, T., Szekely, O., Szekely, P., Ginsburg, A., Asor, R., Resh, R., Tamburu, C., Peres, M. *et al.* (2011). *Soft Matter*, **7**(4), 1512–1523.
- Schrödinger, L. (2015). *The PyMOL Molecular Graphics System, Version 1.8*.
- Schrödinger, LLC (2015). The PyMOL molecular graphics system, version 1.8.
- Svergun, D., Barberato, C. & Koch, M. (1995). *Journal of Applied Crystallography*, **28**(6), 768–773.
- Valentini, E., Kikhney, A. G., Previtali, G., Jeffries, C. M. & Svergun, D. I. (2014). *Nucleic acids research*, **43**(D1), D357–D363.