User Manual

TEM Simulation of Thermal Diffuse Scattering (TDS)

This document is meant to serve as a practical guide to simulate electron diffraction pattern that include diffuse features arising from thermal vibrations. For a general understanding of the principles of the simulation refer to the Supplementary Information of this paper and previous publications from A. S. Eggeman. The simulation is written in C++ and employs NVIDIA's parallel computing architecture CUDA, i.e. a CUDA enabled graphic card from NVIDIA is required to run the simulation. The C++/CUDA source code is provided with examples for TIPS-P, diF-TESADT, C8-BTBT. We strongly suggest starting to work with the one of the provided examples as a quick change of the input parameters will provide a good feeling of how the simulation works.

The first version of this program is based on the Kirkland multislice code for calculating electron waves from the crystal potential (Kirkland, E. J. 'Advanced Computing in Electron Microscopy', Plenum Press, New York, 1998). If you use or alter this source code, please reference:

- Eggeman, A. S et al. Ultramicroscopy, 134 (2013), pp 44-47.
- Illig, S. et al. Nature Communication (2015)

The program is free software and falls under the GNU General Public License, see Disclaimer at the end of this document.

Overview

The simulation prompts the user for an input string that identifies the input files. For example, enter *c8btbt* to work with the provided files that initialize a C8BTBT simulation. Subsequently, the program expects two files *c8btbt_structure* and *c8tbtb_input* that contain all required information. The structure file is a crystallographic information file (.cif file) that can be loaded by standard crystal viewers such as CrystalMaker or Mercury. The input file provides all parameters needed by the simulation. By default the source code simulates [001] diffraction pattern. It is described how to simulate other major zone axis including [100] and [010]. Arbitrary axis' can be simulated but this requires changes in the program as this procedure that has not yet been automated. Here, a short guide is provided how to create the structure and input file for a new material without the need to change any line of source code.

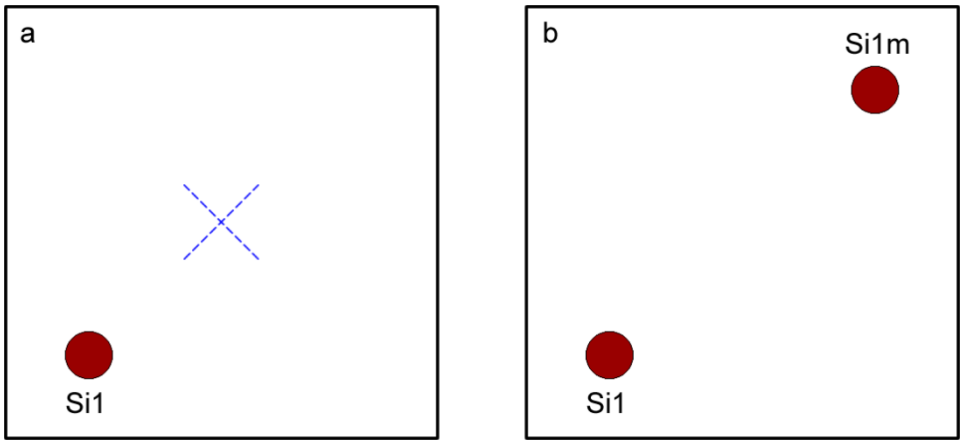
Structure File

The structure file is minimalistic cif file that includes all parameters to describe the ideal crystal structure, i.e. without thermal displacements. Any new structure file has to exactly follow the structure of one of the example files as the parameters are expected in a certain order. This includes:

- Row 1: File name or name of the material (not important for the simulation)
- Row 2: Empty
- Row 3-8: Lattice parameter in the order a, b, c (in Å), alpha, beta, gamma (in degree)
- Row 9: Volume (in $Å^3$) – the simulation compares this value against the volume calculated from the lattice parameter. If the difference exceeds $1Å^3$ the simulation will output a warning. This is to ensure that the parameters have been entered correctly.
- Row 10: Empty
- Row 11-16: These rows are ignored by the simulation but required by cif file readers to understand the order in which atomic parameter are supplied.
- Row 17-end: Every row contains parameters for one atom. The parameters need to be tabulator separated and in the following order:
    - Atomic label (ignored by simulation – only for cif reader)
    - Chemical symbol
    - Fractional coordinate in x
    - Fractional coordinate in y
    - Fractional coordinate in z

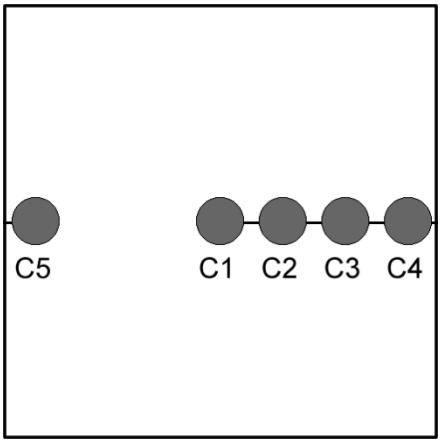
The documentation on the cif file format is good resource if any element of the structure file remains unclear, e.g. what *fractional coordinate* refers to.

In general, cif files of crystal structures that incorporate any symmetry (such as point symmetric structures) only describe the symmetry independent atoms. Any symmetry needs to be resolved, meaning that every atom needs to be described independently, see Tutorial Figure 1.



TDS Tutorial Figure 1: Resolving symmetries. a, Generally, cif files describe only symmetry independent atoms (Si1) and the symmetry of the crystal structure (indicated by the blue cross). b, The structure file for the TDS simulation requires every atom to be described separately (Si1 and Si1m). Atoms that follow from resolved symmetries are often label with an additional m (*mirrored*) in the example files.

After resolving possible symmetries it is crucial to ensure that the fractional coordinates describe connected molecules. For example, consider a fictional molecule that consists of 5 carbon atoms which are distributed in the unit cell as shown in TDS Tutorial Figure 2. The fractional coordinates (in horizontal direction) of the carbons might read as 0.5, 0.65, 0.8, 0.95 and 0.1. For the TDS simulation to incorporate the thermal displacements correctly it is required to provide the fractional coordinates in a way that it logically describes connected molecules. This can be achieved by adding or subtracting an integer value (usually +/-1) to any of the provided coordinates. In the example given in TDS Tutorial Figure 2 the coordinates could be written as -0.5, -0.35, -0.2, -0.05, 0.1 or alternatively 0.5, 0.65, 0.8, 0.95, 1.1. Be aware that a cif file viewer will display the same structure regardless whether an integer value has been added to any of the fractional coordinates as the atomic positions are usually translated back to their unit cell.



TDS Tutorial Figure 2: Connected molecules. It is important that the fractional coordinates in the structure file describe logically connected molecules.

To simulate the other major zone axis [100] and [010] without changes in the source it is possible to *flip* the structure, e.g. to simulate a [100] diffraction pattern the structure can be adjusted so that the former x coordinate falls on z (adjust lattice parameters respectively).

To speed up the simulation it is common to neglect hydrogen atoms as they hardly scatter electrons (scattering intensity scales with the square of the atomic number). Additionally, they are usually relative uniformly distributed among the unit cell.

Always inspect the structure file with a cif file viewer to make sure that everything is described correctly. If the structure file can't be loaded by the viewer it is likely that the syntax is wrong which will probably also cause problems during TDS simulations.

Further adjustments of the structure file are sometimes required to fit structures with gamma ≠ 90° in a rectangular matrix for CUDA's FFT algorithm. This process is explained in the input file section.

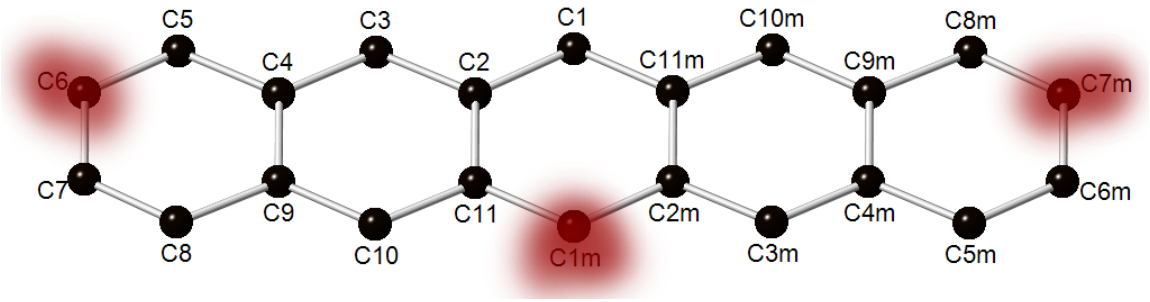
Input File

The input file is a simple text file that contains a number of parameters describing everything from the instrumental setup over the vibrational model to the output file. Exactly as in the structure file these parameters are expected by the simulation in the order they are listed in the example files. The input file is somewhat self-explanatory, with a description of each parameter before the actual value. Parameters include:

- **Output file suffix:** The output file will be stored as *input string* + *underscore* + *output file suffix* + *.txt* (TextImage file). For example, if the input string is *c8btbt* and the output file *longaxis20_5runs*, the resulting output will be stored as *c8btbt_longaxis20_5runs.txt*.
- **Averaging:** Number of times the simulation will run. The resulting output pattern will be the average of all runs. This makes the output pattern smoother but obviously increases simulation time.
- **Thickness scan:** Either *Yes* or *No* is expected here. In a thickness scan an output pattern is written not only after the incident electron wave penetrated through the entire crystal of the chosen thickness but after every slice (see *multislice* simulation). This option results in a number of output patterns suited to refine the experimental film thickness by comparison of the simulated patterns with the experimental diffraction pattern. Averaging will be set to 1 during a thickness scan, i.e. the simulation only runs once.
- **Acceleration voltage:** TEM acceleration voltage in keV.
- **Size of simulated super cell:** Two values are expected setting the size of the super cell in x and y direction (in-plane). For example 8 and 12 will create a 8x12xT super cell whereas T is the film thickness in molecular layers (next parameter).
- **Thickness in molecular layers:** Size of the simulated super cell in z direction (out-of-plane). A value of 20 will create a crystal which is 20 molecular layers thick.
- **Slice thickness:** The multislice simulation works in a way that is cuts the unit cell in slices and subsequently transmits the electron beam through each slice. The thickness of each slice will determine the total number of slices. Example: The film thickness is set to 20 molecular layers and the crystal lattice parameter in z is 16.835Å (length of unit cell in z). The simulated film thickness is therefore 20 x 16.835Å = 336.7Å. If a slice thickness of 10Å is chosen the program will cut the super cell in 34 slices of 9.9Å as it rounds the selected value to ensure slices of equal thickness. Thinner slices will better reproduce dynamic scattering effects but run slower. Code of good practice is to set the slice thickness to the thickness of one molecular layer (length of unit cell in z, e.g. 16.835Å in TIPS-P).
- **Random Noise:** This is usually set to zero. A non-zero value *a* will apply random displacements to all coordinates (x, y, z) of all atoms. The displacements are randomly chosen from a Gaussian distribution with *sigma = a* (definition of the amplitude in the main text of the paper) and superimposed on other vibrational modes defined later.
- **Debye Waller Factor (DWF):** This should always be zero. Be aware what a DWF is and ask yourself to which extend it makes sense to define such a factor if particular thermal modes are explicitly simulated. If, for whatever reason, a DWF shall be applied, refer to the source code how it is implemented in the simulation as there are multiple definitions.
- **Size of the output pattern (array size):** Pixel size of the resulting output pattern. Two numbers (width and length) are expected. Note that high frequencies (edges of the pattern) are cut off during simulation, meaning that the effective output pattern size is smaller.

Ideally the array size should be selected as a power of two (e.g. 512, 1024, 2048). To start with a good size is 512x512 as larger array sizes will strongly increase simulation time.

- **Defining Coordinate Systems:** Here it is possible to describe one or several coordinate systems that will allow specifying the direction of thermal displacements. Each coordinate system is defined by the keyword *Coordinate System:* followed by three numbers separated by commas. Each number refers to an atom in the structure file whereas the first atom (starting in the structure file from row 17) is indexed with 0, i.e. if the structure file lists 25 atoms, indices run from 0 to 24. The first two atoms define the x axis in the new coordinate system (in the example files this has generally been chosen to correspond to the long axis of the conjugated cores). The third atom defines a plane with the other two. The y axis (corresponding to the short axis in the example files) lies on that plane but perpendicular to x. Finally the z axis (direction of the π-π stacking in the example files) lies perpendicular to x and y. This results in a Cartesian coordinate system x, y, z. Several such systems can be defined, e.g. for each molecule with different orientation in the unit cell. Tutorial Figure 3 visualizes how a coordinate system can be defined with x corresponding to the long-axis of a pentacene molecule, y to its short axis and z in direction of the π-π stacking by providing the indices of atoms C6, C7m and C1m in that order.
- **Describing vibrational modes (phonons):** An unlimited number of phonons can be described, each beginning with the keyword *Phonon:* followed by 7 comma-separated parameters:
    - **rot/trans:** Specifying whether the vibrational mode is a libration (around the specified direction) or translation (in the specified direction).
    - **Sigma:** Amplitude of the vibrational mode. The thermal displacements are randomly drawn from a Gaussian distribution with a sigma of the selected value (definition of *amplitude*).
    - **Group:** Defines the subgroup of atoms the vibrational mode applies to (subgroups are defined in the next paragraph).
    - **Coordinate System:** Index of the coordinate system that is used to describe the direction of the vibrational mode (indices start at 0).
    - **Direction in x, y, z:** Describes the direction of the vibration. E.g. 1, 0, 0 describes a vibration along the x-axis (usually used to describe the long-axis of the molecule). 1, 1, 0 will define a direction that lies in the x-y plane and that describes a 45˚ angle with both axis' etc.
- **Groups of Interest:** This assigns every atom a number that defines its group. The simulation expects one number in a separate row for every atom in the unit cell. For example, consider a crystal contains two molecules of different directions A and B. Vibrations in different directions shall be assigned to A and B. However, only the conjugated cores are supposed to vibrate as rigid units with the side chains being rigid (general approximation if the side chains do not follow the vibrations of the molecular core). It this scenario it is possible to assign the index 0 to all side chain atoms, 1 to the conjugated core atoms of molecule A and 2 to all conjugated core atoms of molecule B. When describing the vibrational modes (previous step) assign 1 or 2 to *Group* according to which atoms the vibrational mode shall be applied to.



Tutorial Figure 3: A Cartesian coordinate system that describes the long axis of the molecule as x, its short axis as y and z as the π-π stacking direction can be defined through C6, C7m, C1m. C6 and C7m defines x, C1m defines with C6 and C7m a plane in which y lies perpendicular to x and finally z lies perpendicular to that plane. Note that the input file expects the indices of these atoms (e.g. 7) rather than their labels (e.g. C6).

Confirm the vibrational model

The super cell that is used for simulation will be written as a .cif file in the early phase of the simulation process. Always load this file with a standard crystal viewer and confirm that the vibrational model has been written as expected, e.g. that vibrational modes are applied correctly.

More complex vibrations

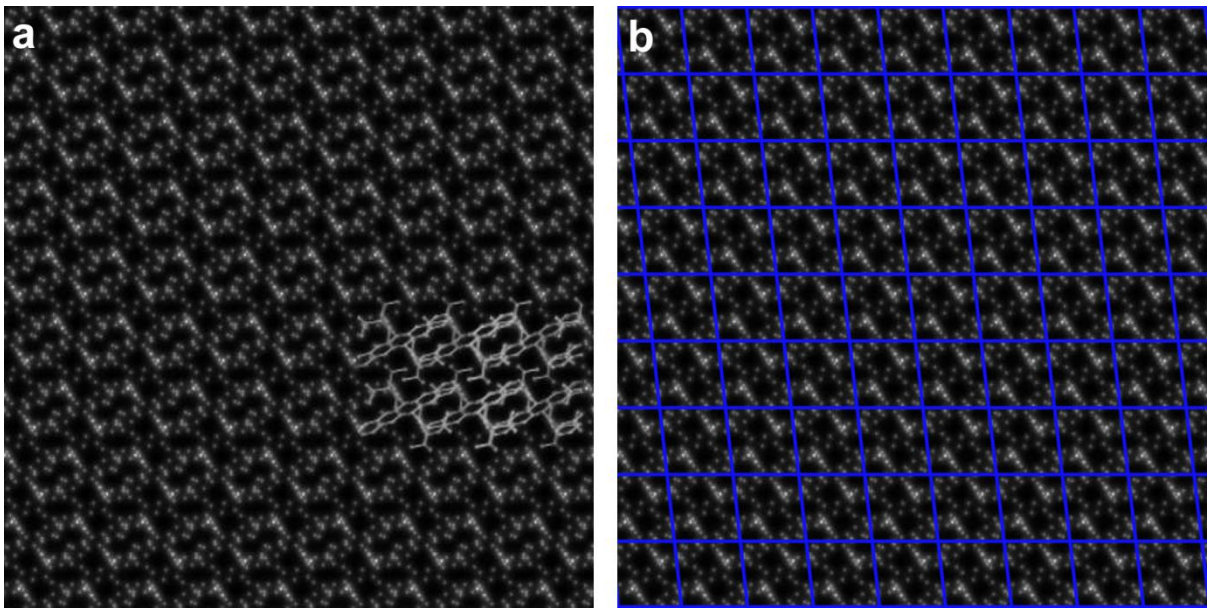
More complex vibrations such a breathing mode of a molecular fragment requires a direct description of the atomic displacements in the source code as such modes have not (yet) been standardized and made available by simply setting input file parameters.

Size of simulated cell and orthogonal matrix

The FFT algorithm that is used in the simulation imposes a few requirements on the super cell size. Let the unit cell parameters be a, b, c, α, β, γ and the parameters of the super cell a', b', c', α', β', γ'. For example a 4x4x10 super cell might for example be described by parameters a'=4a, b'=4b, c'=10c, α'=α, β'=β, γ'=γ. The FFT algorithm requires that a super cell is found with γ'=90˚ (angle in a-b plane) and a'=b'. Often it is not possible to meet these requirements exactly in which case a close approximation is sufficient. Take a look at the examples provided below:

- Assuming a structure with γ=90˚ and a=8Å and b=16Å any super cell i1 x i2 x i3 with i1=2*i2 satisfies the requirements exactly.
- Assuming a structure with γ=90˚ and a=6.8Å and b=15.8Å a super cell i1 x i2 x i3 with i1=21 and i2=9 might be a good choice as 21*6.8Å=142,8Å is approximately 9*15.8Å=142,2Å. Note that the resulting structure will be slightly distorted (by approx. 0.5%) which can be corrected for manually after simulation.
- For a unit cell with γ≠90˚ it is important to choose parameters so that the new unit cell (super cell) satisfies γ'=90˚ and that correctly allows building large crystals upon repetition of the unit cell. For example, the TIPS-P unit cell describes an angle γ that shifts each molecular row by an amount so that after 9 repetitions the molecule corresponds to its starting position, see Tutorial Figure 4. For that reason a 9x9 or 18x18 super cell is chosen for simulations. The angle γ of the structure file can also be slightly adjusted to satisfy this condition without having a strong impact on the simulation.
- If the Bragg reflections in the simulated diffraction pattern smear out it is an indication that the requirements above are not satisfied sufficiently. This step requires some experience

which is why it might be a good idea to start working with the structures described in the example files and to compare different solutions with the chosen approximations.



Tutorial Figure 4: a, 9x9 phase grating of the TIPS-P crystal structure (see referenced publication by A. S. Eggeman). A few molecules have been overlaid with rendered ball and stick models to visualize the positioning of the TIPS-P molecules. b, Any 9x9 repeat of the unit cell can be described by a new super cell structure that satisfies the γ'=90˚ requirement.

Super cell size and size of simulated diffraction pattern (array size)

Both, the super cell size and the array size have an effect on simulation time. For the first simulations and initial refinements an array size of 512x512 should suffice. Note the relationship between super cell size and array size. Given a super cell size of 15x20xi3 and an array size of 512x512 the distance between (000) and its adjacent reflection (100) will be 15 pixels. The distance between any two adjacent reflections in y (e.g. (000) and (010) will be 20 pixels. With the higher frequencies being cut off by the simulation (this creates the black disk in the simulated patterns) reflections might be visible to the 10^{th} order but not more. If higher order reflections shall be visible it is possible to either increase array size or to decrease the super cell size. For a super cell of 8x8xi3 only 7 pixels lie between Bragg reflections, meaning that even in a 512x512 pattern Bragg reflections up to a higher order are visible. However, a decrease in unit cell size worsens statistics and might be limited by the FFT requirements described in the previous paragraph. The choice between these parameters is therefore a trade-off between statistics, high spatial resolution and simulation time.

Disclaimer

The computer code and or data in this file is provided for demonstration purposes only with no guarantee or warranty of any kind that it is correct or produces correct results. By using the code and or data in this file the user agrees to accept all risks and liabilities associated with the source code and/or data. The source code and/or data in this file may be copied and used for non-commercial academic or research purposes only, provided that this notice is included. This file or any portion of it may not be resold, rented or distributed without the written permission of the author.

The original code from Kirkland has been significantly altered for use under GPU conditions and for calculating thermal diffuse scattering dynamically. Please reference:

- Eggeman, A. S et al. Ultramicroscopy, 134 (2013), pp 44-47.
- Illig, S. et al. Nature Communication (2015)