



2019 - Annual School on
Neutron Diffraction Data
Treatment using the FullProf
Suite

Introduction to the FAULTS program

M. Reynaud
J. Rodríguez-Carvajal
M. Casas-Cabanas



Program

- | | |
|-------------|---|
| 9:00-9:30 | Introduction to FAULTS |
| 9:30-10:15 | Example 01: Stacking faults in graphite.
Build the structural model in a FAULTS input file. Export the structural model to VESTA. Simulate an XRD pattern. |
| 10:45-12:00 | Example 02: Stacking faults in LiNiO_2
Build the structural model in a FAULTS input file. Simulate an XRD pattern. |
| 14:00-16:00 | Example 02: Stacking faults in LiNiO_2 (continue)
Refine an XRD pattern. Add a secondary phase. |
| 16:30-18:30 | Example 03: Stacking faults in Li_2MnO_3
Build the structural model of a monoclinic structure in a FAULTS input file.
Simulate an XRD pattern and electron diffraction patterns. Generate the Bragg positions of an average cell. |

Example 01: Stacking faults in graphite

In the graphite structure, the carbon atoms are ordered in 6-membered rings in the layers. The layer stacking can repeat itself at intervals of either two or three layers (Figure 1). This special form of polymorphism is called polytypism. In these polytypic structures, the first one (a) gives a hexagonal unit cell while the second one (b) gives a rhombohedral. The structures are thus labelled as the 2H- and the 3R-polytypes of graphite, respectively. Graphite 2H consists of two “graphene” layers (Figure 2) at positions A and B (Figure 1a), whereas graphite 3R consists of three “graphene” layers at positions A, B and C of a triangular lattice (Figure 1b).

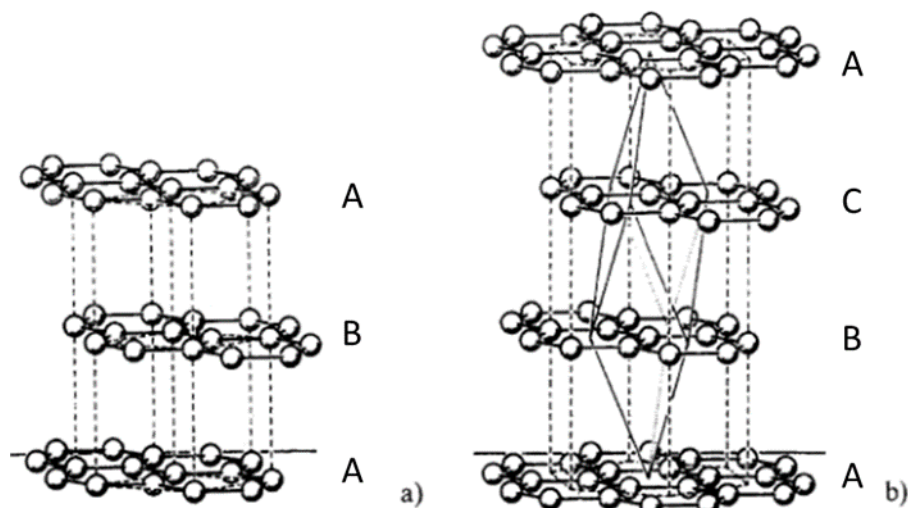


Figure 1: Polytypes of graphite a) 2H; b) 3R

(adapted from W. Borchardt-Ott, *Crystallography*, Springer, 1995, ISBN: 978-3-540-59478-9, DOI: 10.1007/978-3-642-57754-3 I.)

Crystallographic information of the two polytypes of graphite:

2H polytype			
Space group: $P 6_3/m m c$			
Unit cell parameters:			
$a = 2.464 \text{ \AA}$	$b = 2.464 \text{ \AA}$	$c = 6.711 \text{ \AA}$	$\gamma = 120^\circ$
Atomic positions:			
	x/a	y/b	z/c
C1	0	0	0.25
C2	1/3	2/3	0.25

3R polytype:			
Space group: $R\bar{3}m$			
Unit cell parameters:			
$a = 3.635 \text{ \AA}$	$b = 3.635 \text{ \AA}$	$c = 3.635 \text{ \AA}$	$\alpha = \beta = \gamma = 39.49^\circ$
Atomic positions:			
	x/a	y/b	z/c
C1	0.164	0.164	0.164

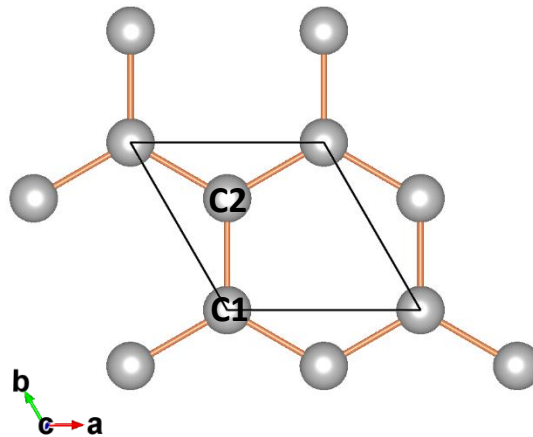


Figure 2: In-plane view of a graphene layer along the c axis in the 2H hexagonal unit cell.

Exercise: graphite polytypes

- 1.1. a) Build the structural model of both polytypes 2H and 3R using the program VESTA.
 b) Using VESTA's tools, transform the structural description of the 3R polytype from the rhombohedral setting (setting 2 of the space group $R\bar{3}m$) to the hexagonal setting (setting 1 of the space group $R\bar{3}m$). Tip: to do so, go to the menu *Edit* → *Edit data* → *Unit cell*, select the setting of your choice in the right box (check that the option “*Update structure parameters to keep 3D geometry*” is selected in the pull-down menu on the right side of the window), and confirm with the buttons *Apply* and *OK*: the structural description is automatically transformed by VESTA.
 c) Looking at the structures of both polytypes, imagine how they could be described using layers and stacking vectors (several descriptions are possible for each polytype).
- 1.2. Build the simulation input file for the 2H polytype using only one type of layer (structure described with several transition vectors). Check your FAULTS model using VESTA. Simulate with FAULTS the XRD pattern of the 2H graphite, and compare it with the XRD pattern of the 2H graphite simulated with FullProf.
- 1.3. Build the simulation input file for the 3R polytype using only one type of layer (structure described with several transition vectors). Check your FAULTS model using VESTA. Simulate with FAULTS the XRD pattern of the 3R graphite, and compare it with the XRD pattern of the 3R graphite simulated with FullProf.
- 1.4. Using the same FAULTS cell, build the input file to describe a 3R structure including 40% of 2H stacking. Check your FAULTS model using VESTA, and simulate the XRD pattern of this intergrowth of graphite 3R and graphite 2H.

Example 02: Stacking faults in LiNiO_2

Lithium nickel oxide has been intensively studied as a positive electrode material in Li-ion batteries. As other lithium transition metal oxides, it crystallizes in an O3-type layered structure, with an ABCABC oxygen stacking sequence, and lithium ions located in the octahedral sites of the interlayer spaces (see Figure 3). O1 stacking faults that represent a break in the normal stacking sequence of the structure, with a local ABAB oxygen stacking sequence, can appear at low lithium concentrations.

NOTE: For simplicity, we will consider that the lithium concentration is the same in all layers (see FAULTS manual for the full example with low lithium concentration in O1 motifs).

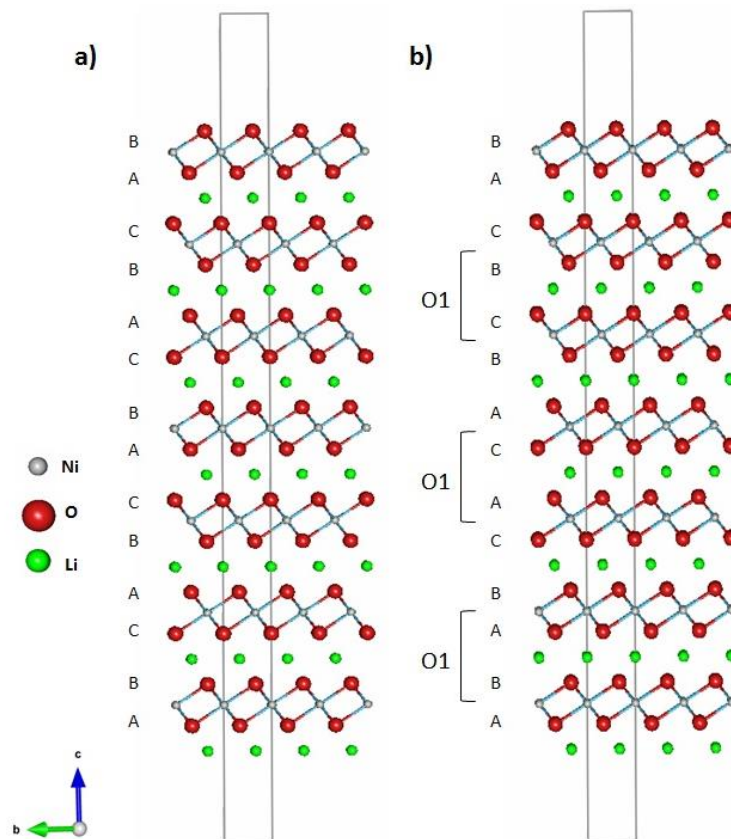


Figure 3: Structure of LiMO_2 a) ideal, b) faulted.

Crystallographic information of LiNiO_2 :

LiNiO_2			
<i>Space group: $R\bar{3}m$</i>			
<i>Unit cell parameters:</i>			
$a = 2.8769 \text{ \AA}$	$b = 2.8769 \text{ \AA}$	$c = 14.198 \text{ \AA}$	$\gamma = 120^\circ$
<i>Atomic positions:</i>			
	x/a	y/b	z/c
<i>Li</i>	0.0	0.0	0.5
<i>Ni</i>	0	0	0
<i>O</i>	0	0	0.2580

Exercise:

- Build the structural model of LiNiO_2 . Check your FAULTS model using VESTA. Simulate the XRD pattern of ideal LiNiO_2 . Tip: you can use the atomic positions that result from transforming the cell to $P1$ space group. This can be easily done with VESTA: go to the menu Edit → Edit data → Unit cell → remove symmetry, and confirm with the buttons *Apply* and *OK*.
- Build a structural model for LiNiO_2 with stacking faults. Check your FAULTS model using VESTA. Explore the effect of having varying amounts of O1 type stacking faults in LiNiO_2 .
- Build the input file to refine LiNiO_2 with stacking faults and NiO impurities. You need to include a regular background, the NiO background file and select the refinable parameters. After the refinement process analyse your results and, if needed, run the refinement again using the FILENAME_new.flts file.

Example 03: Stacking faults in Li_2MnO_3

Li_2MnO_3 , which can also be described as $\text{Li}[\text{Li}_{1/3}\text{Mn}_{2/3}]\text{O}_2$ to better take into account its layered structure, can be indexed with the $C2/m$ space group. The XRD pattern usually exhibits superstructure peaks in the range $2\theta = 20\text{--}35^\circ$ that indicate that the Li/Mn atoms of the mixed layer are arranged following a “honeycomb” ordering scheme (see Figure 4).

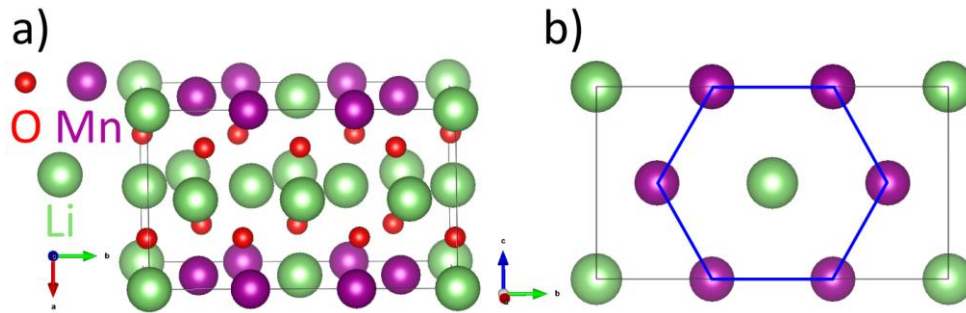


Figure 4: a) Li_2MnO_3 $C2/m$ unit cell and b) view of the Li/Mn layers where the honeycomb arrangement can be observed.

This extra ordering leads to three stacking possibilities as shown in Figure 5 and therefore this type of material is prone to exhibit stacking faults. The concatenation of such stacking faults generates a characteristic asymmetric peak broadening as a consequence of the ordering loss (Warren fall).

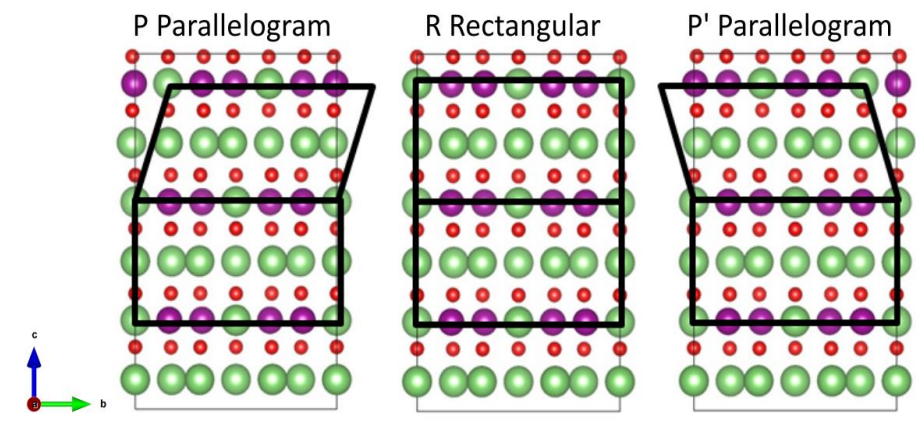


Figure 5: Possible layer transitions in the crystal structure of Li_2MnO_3 . R shows the ideal stacking while P and P' represent stacking faults.

In the following exercise we will analyse this peak broadening. Additionally, the effect this disorder causes to the ED patterns will also be evaluated.

Li₂MnO₃			
<i>Space group: C 2/m</i>			
<i>Unit cell parameters:</i>			
$a = 4.9292 \text{ \AA}$	$b = 8.5315 \text{ \AA}$	$c = 5.0250 \text{ \AA}$	$\beta = 109.337^\circ$
<i>Atomic positions:</i>			
	x/a	y/b	z/c
<i>Li 2b</i>	0.000	0.000	0.000
<i>Li 2c</i>	0.500	0.000	0.500
<i>Li 4h</i>	0.000	0.847	0.500
<i>Mn 4g</i>	0.000	0.328	0.000
<i>O 8j</i>	0.747	0.811	0.751
<i>O 4i</i>	0.262	0.000	0.755

Exercise:

a) Build the .flts file and simulate the powder diffraction pattern of Li₂MnO₃ using the provided .cif file. NOTE: In FAULTS program the stacking direction must be perpendicular to the ab plane, which means that β must be 90° and hence a new cell c parameter (perpendicular to both a and b) has to be calculated. This will affect both the description of atomic positions as well as that of layer transitions.

b) Explore the effect of varying degrees of stacking faults in Li₂MnO₃.

c) Generate ED patterns with samples with different degree of stacking faults. The files created by FAULTS for the simulated ED patterns are unsigned 16-bit images of size 256x256 with a *.sadb extension. They can be opened with a specific software such as ImageJ (available at: <https://imagej.nih.gov/ij/download.html>)

Important remarks to correctly run FAULTS

- Carefully read the messages written by FAULTS in the command window and in the output file .out. Most of the errors can be identified from these messages!
- Tabs are not allowed in the input files. They can generate unpredictable errors! Only spaces should be used. Working with editors such as Notepad ++ can help us to find these mistakes.
- In Notepad ++ you can select and edit several lines at the same time by pressing *alt* while selecting with the mouse. This can be helpful when refining the background parameters, for example.
- All the non-optional sections and keywords have to be present in the .flts input file.
- Contrary to the .pcr input files used in FullProf, a missing refinement code will be not considered as zero but will produce an error, so all the refinement codes have to be present in the .flts input file.
- The combination of the profile parameters U, V, W, X should not lead to a profile function with negative values (no negative Gaussian and/or Lorentzian FWHM). The user can check this by calculating the Gaussian and Lorentzian FWHM (H^2G and HL) with the formula given in page 10 of the FAULTS manual, using the program WinPLOTR (Menu Calculations => I.R.F. (U,V,W,X,Y,Z)) or any spreadsheet.
- The way of working with FAULTS requires especial care of the user. Before starting to do refinements it is advisable to make simulations in order to start with an initial model that is not too far from the experimental diffraction pattern.
- If the structural model is complicated, the first calculations can take time, so the user should make sure that he/she lets the program some minutes before concluding that it is blocked. Keep in mind that the derivatives are calculated numerically by calling two times the total function per free parameter and this calculation may be expensive in CPU-time.
- In refinement mode, at least one parameter should be refined.
- The name of the input files (including the extension) is limited to 29 characters.

Other tips

- Work with Notepad ++ to find errors easily.
 - In notepad remove Auto-Completion options: Settings → Preferences → Auto-Completion → Un-tick options
- See all file extensions:
 - For Windows 10: Open a folder. Go to View tab. Check File name extensions option.

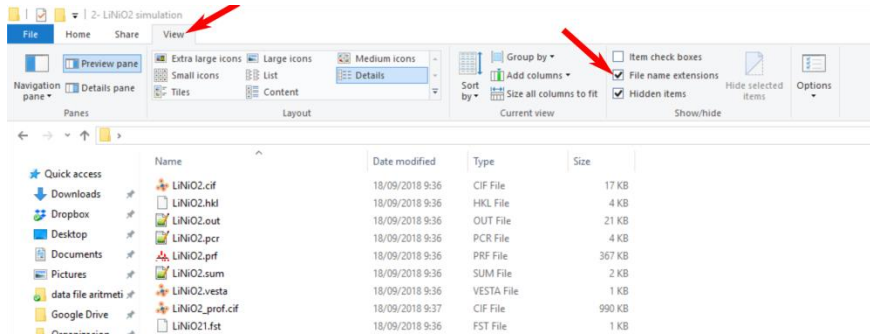


Figure 6. Steps to follow to see all the file extensions.

- To open a console in the folder you want: Shift + right click + Open PowerShell window here.