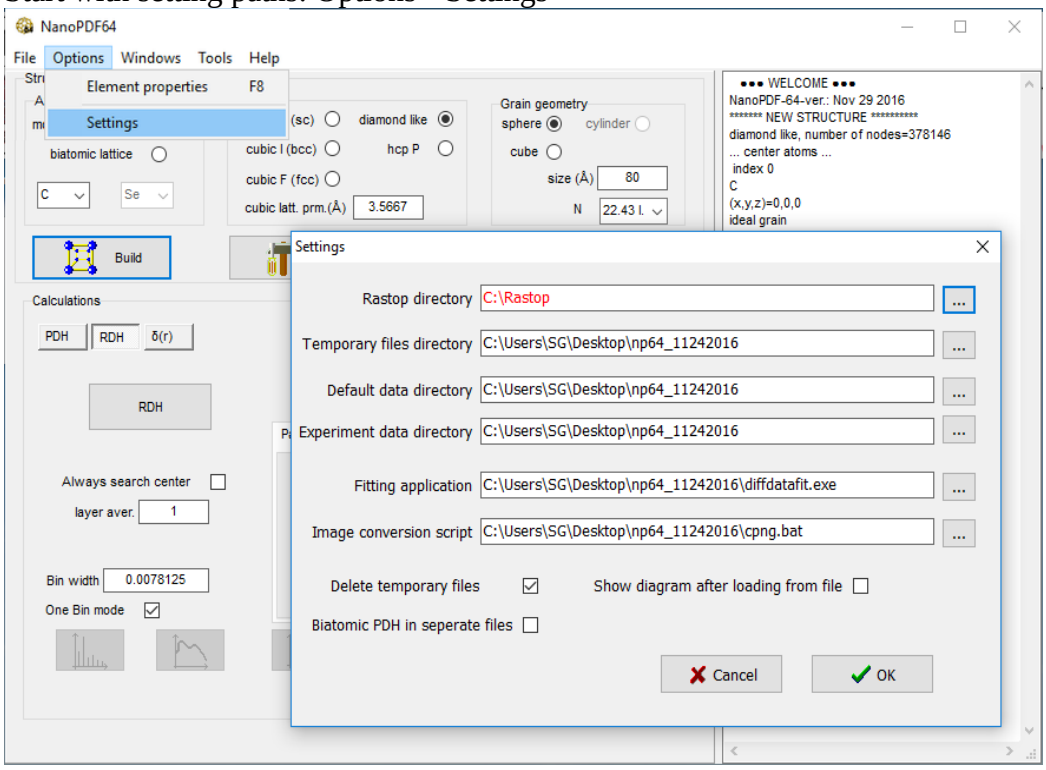
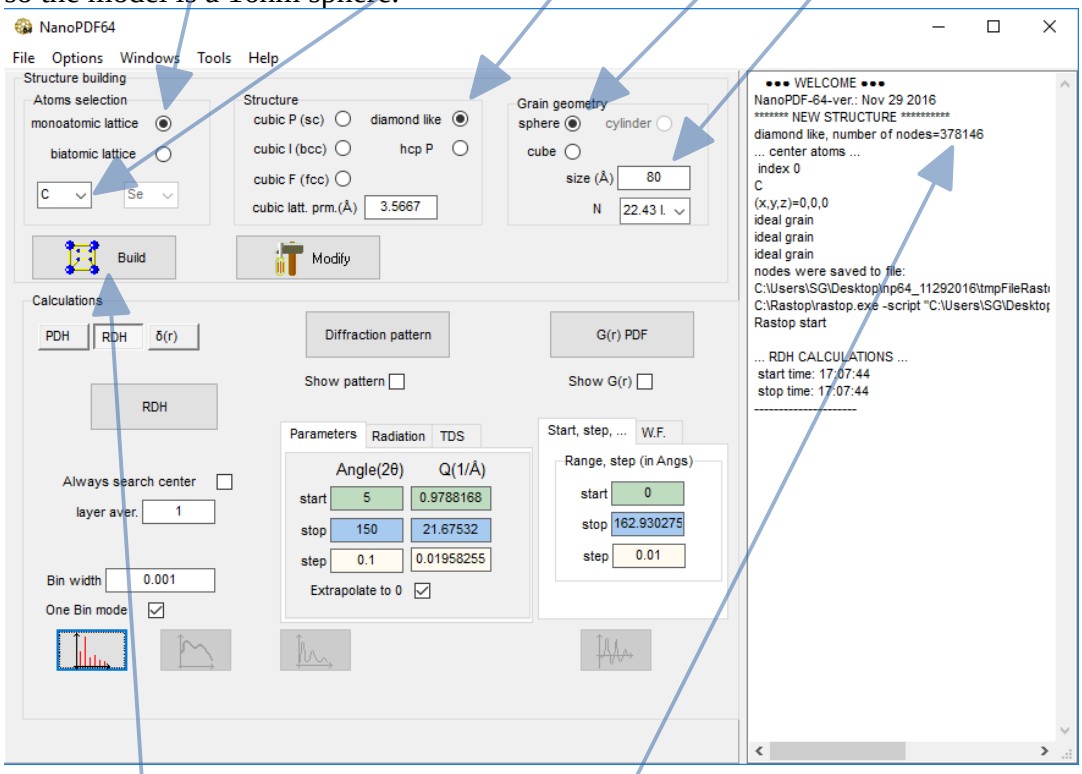
# NanoPDF64
## fitting procedure example

Start with setting paths: Options->Settings

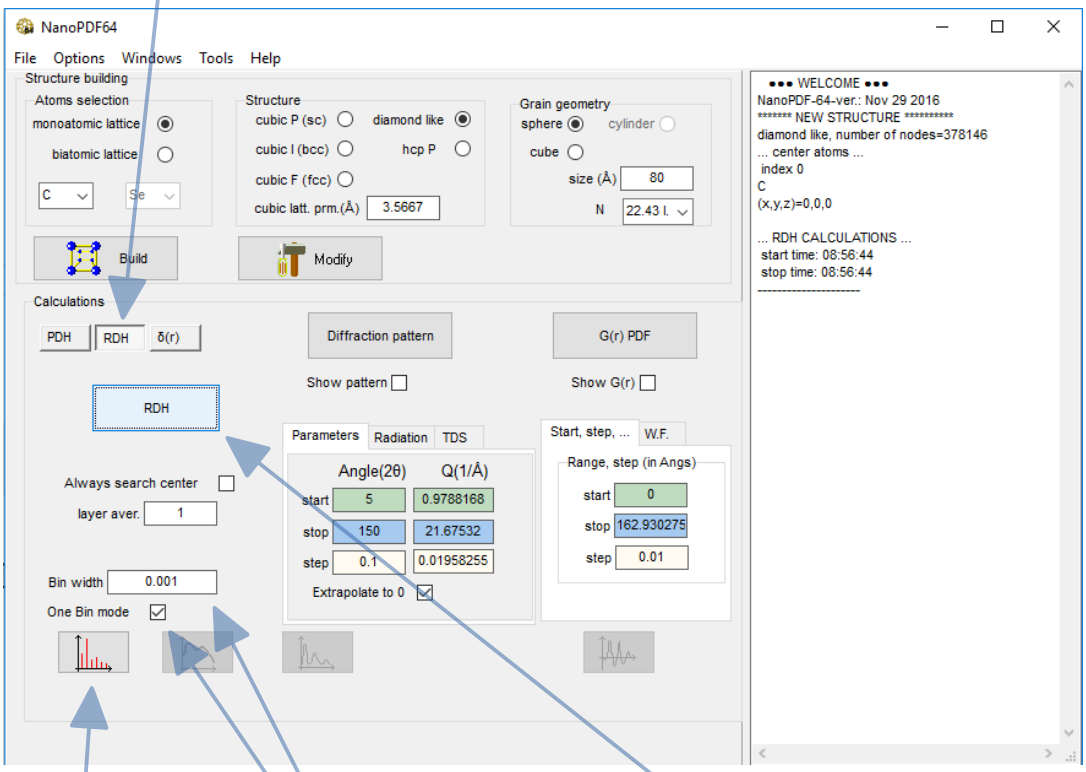



Now build your structure.
Choose: **monoatomic lattice**; Atom: **C**; **diamond like**; **sphere;** size **80Å** - actually this is the radius so the model is a 16nm sphere.



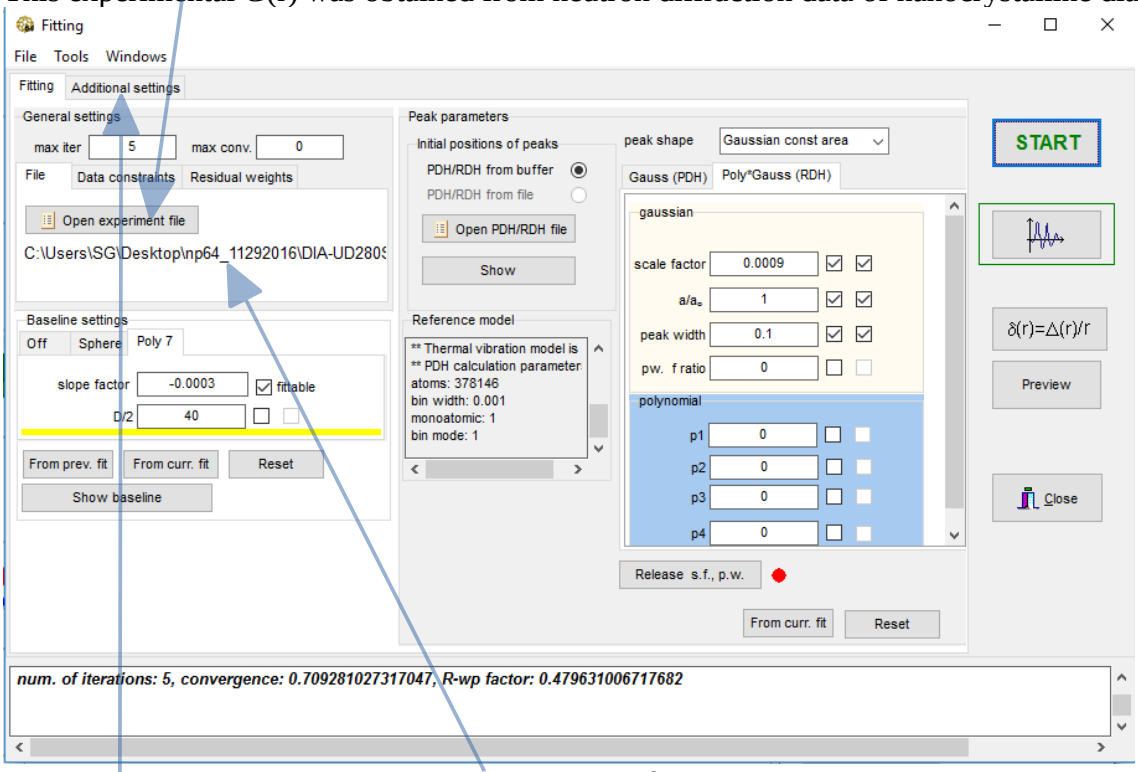

Click on **Build**. At this point you may already look at the model by choosing:
**Tools->Rastop->Ideal**.
However, since the model contains over 370 000 atoms it will be quite a task for Rastop.

Switch to RDH tab.

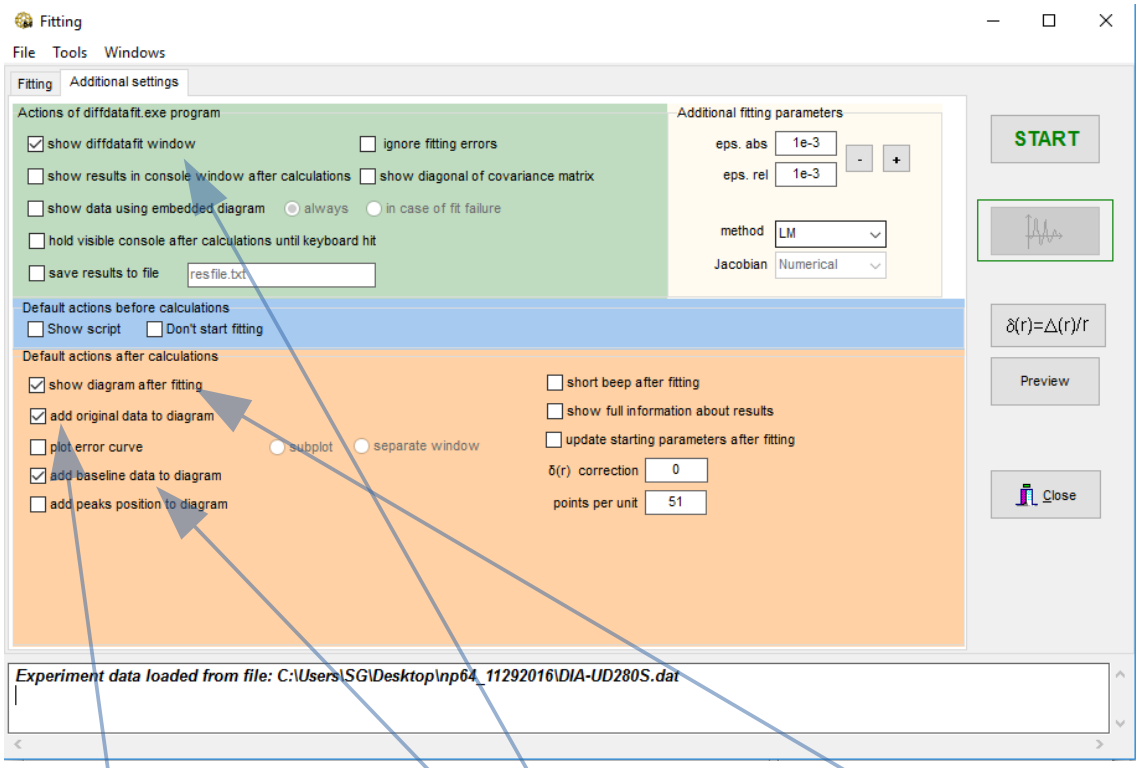



Set **Bin width** to e.g. 0.001 – this is the accuracy of the interatomic distance calculation - and make sure that you are in the **One Bin mode**. Finally click on **RDH** button and press the **Plot** button to have a look at the histogram.

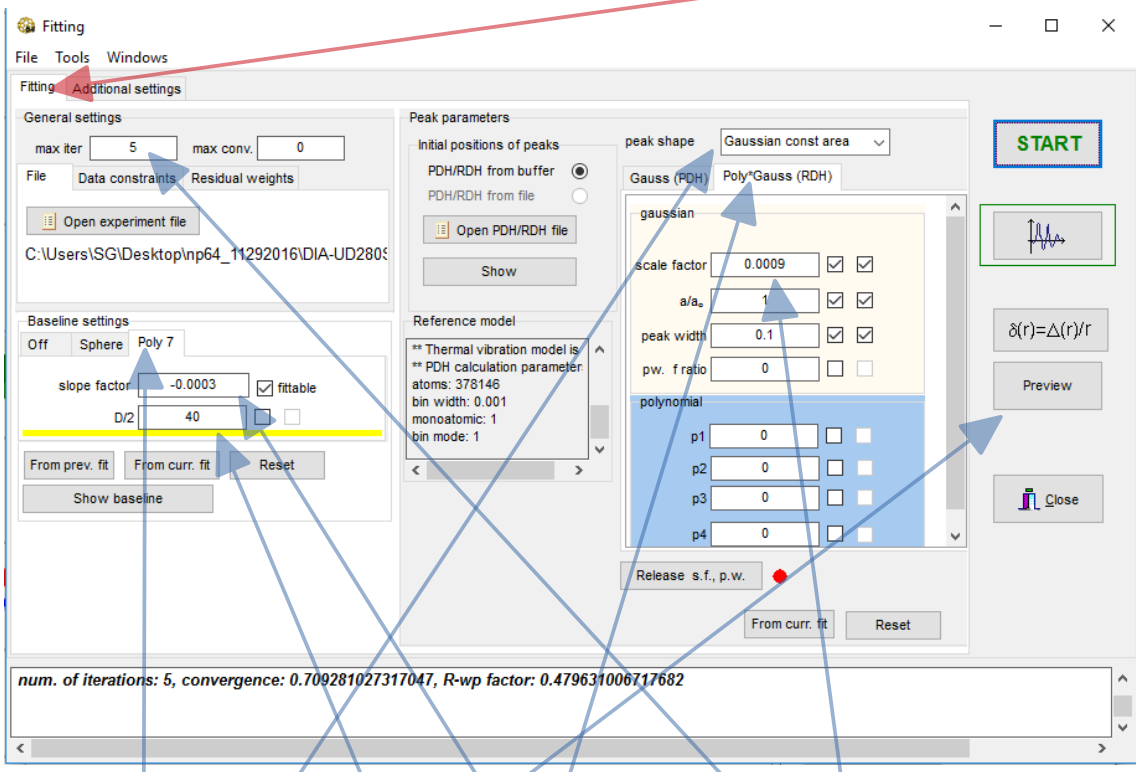
We are now ready to analyze the experimental data. Choose **Tools → Fitting** or press **Ctrl-F**. Start with loading G(r) to be analyzed – use "DIA-UD280S.dat" that comes inside the package. This experimental G(r) was obtained from neutron diffraction data of nanocrystalline diamond



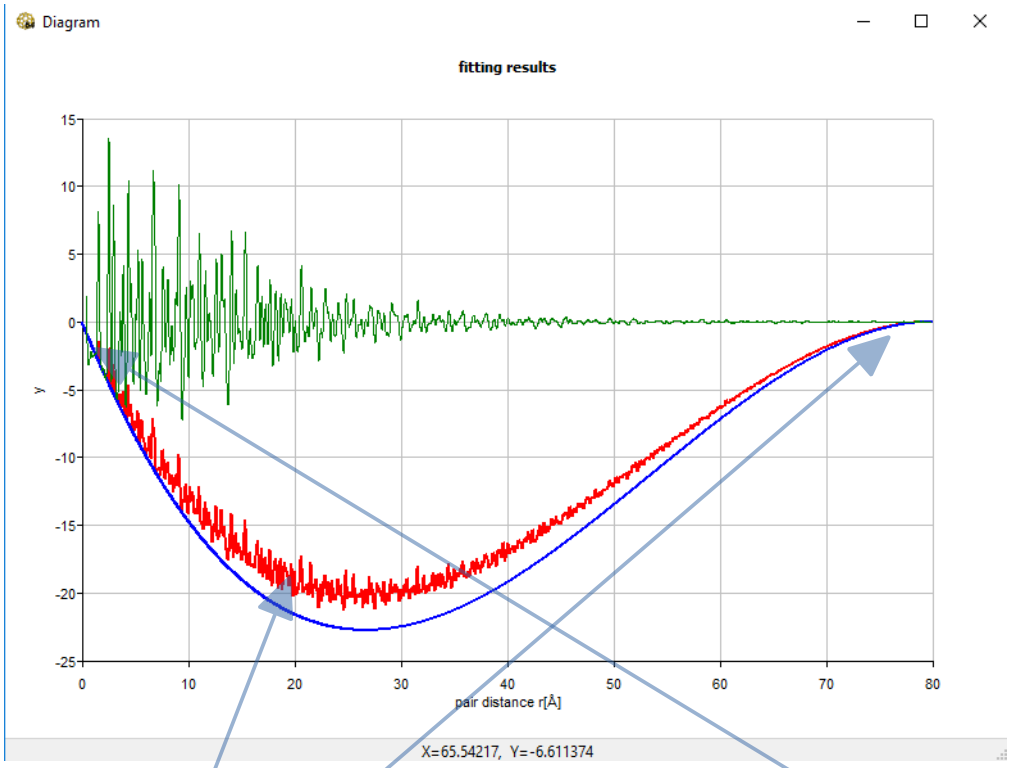

To make sure the data are OK double click on the file name to display the data graph. Now go to **Additional settings** tab:




At this point important setting are: **show diffdatafit window**; **show diagram after fitting**; **add original data to diagram** and **add baseline data to diagram.** Go back to **Fitting** tab.

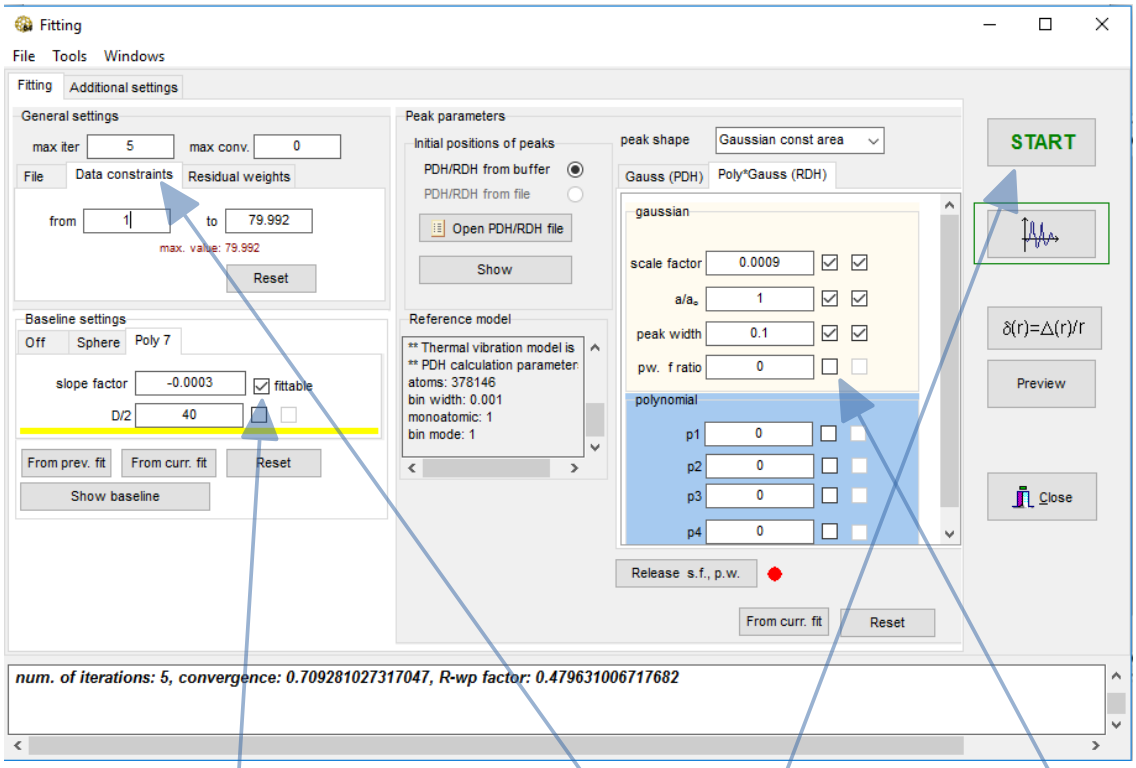



Choose: **Poly7**; **Gaussian const area**; **Poly*Gauss(RDH)**; set **max iter** between 5 and 10. Now we set the initial parameters of theoretical G(r). **Slope factor** controls the slope of the baseline near "0". It should be negative and small. **D/2** is a half of the particle size. **Scale factor** controls the height of the peaks in G(r). Usually it is quite small. **$a/a_0$** adjusts the lattice parameter and **peak width** is self explaining. Use **Preview** button to see if the starting parameters make sense.

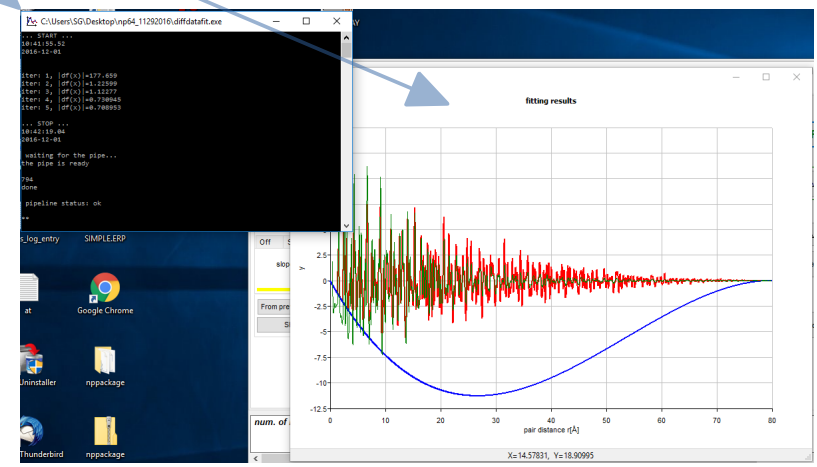



Experimenta data are plotted in **green**. Baseline is **blue**. With **slope factor** you change the baseline slope near "0". With **D/2** you change the point where the G(r) vanishes at large distance. Remember: it is the half of this distance! Theoretical G(r) is **red**. It is the sum of the baseline and all the peaks taken from the RDH histogram you calculated before. The height of the peaks in G(r) is controlled by **scale factor**. Play with **slope factor** and **scale factor** to get the starting theoretical curve reasonably close to experimental one in terms of intensities. Also check the peak positions - **$a/a_0$** - and widths!

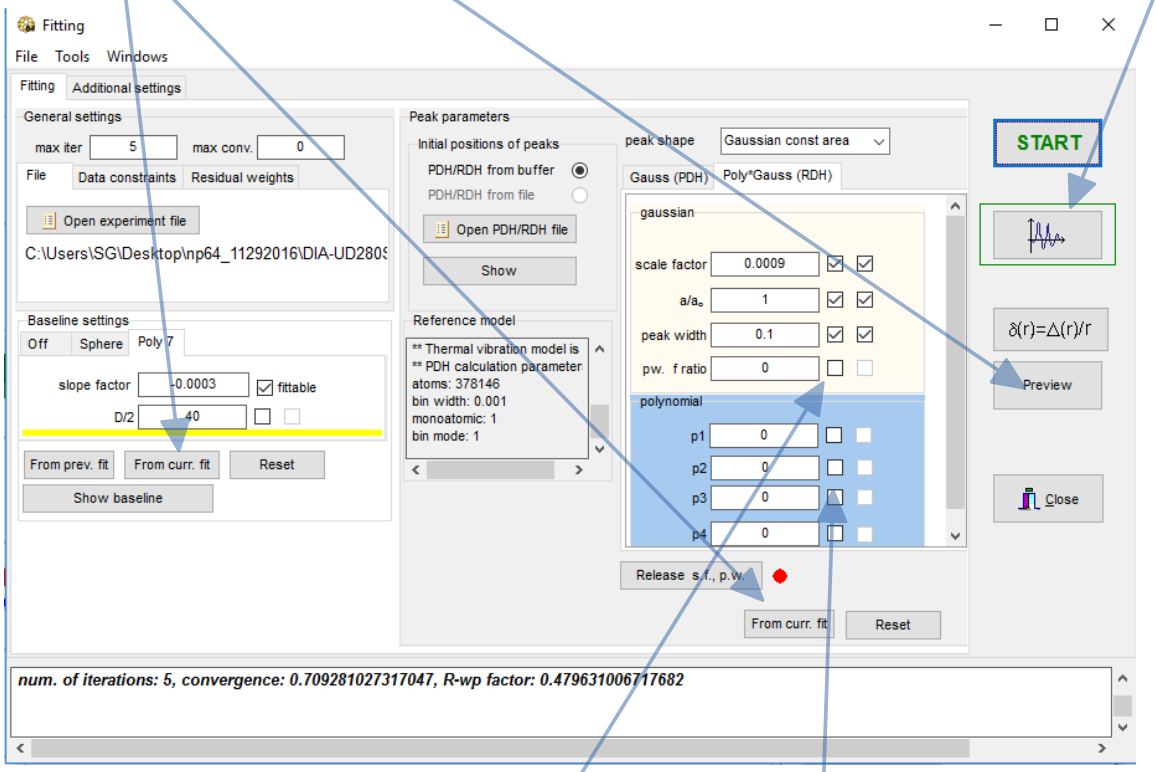
Zoom works by dragging the mouse with **Left** mouse button and **Ctrl** keyboard button pressed. Enlarged graph opens in a new window. You actually only zoom the X-axis. Y–scale is set automatically. If you want to change it press **Right** mouse button on the graph and select the appropriate option from the menu that appears.
Now close the graph windows and go back to **Fitting** window.



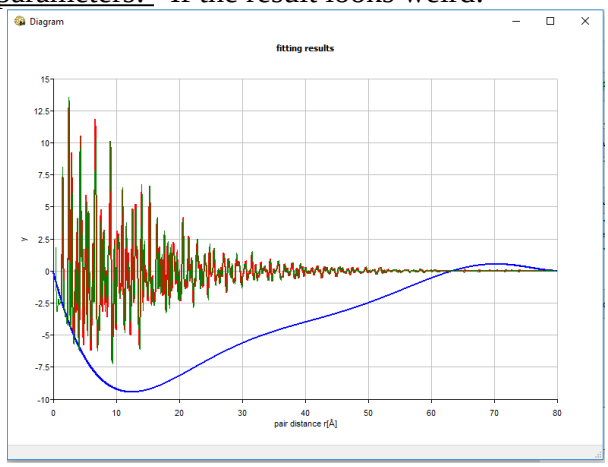

At this point it makes sense to switch to **Data constrains** and limit the range of the analyzed data e.g. disregard the data oscillations at the beginning.
Make the following parameters variable: **slope factor**, **scale factor**, **a/a$_0$**, **peak width**. It is done by checking **the boxes** next to the parameters' values. Make sure all other boxes are **unchecked**.
Doubled boxes near some parameters have special meaning. If you clear the second box, a given parameter is decoupled for all peaks in G(r). If you e.g. decouple intensities, the relative heights of the peaks will not follow the structure geometry (actual distance multiplicity) but will be fitted independently. Usually the fitting procedure goes nuts with such setting. Make sure both boxes are checked for parameters to be varied.
In order to start least squares fitting click on "**START**".
The **console window** will appear showing the calculation progress. When it is over the result will be displayed in the **graph**.



At this point there exist two sets of parameters: those initially set and those calculated by least squares. The **Preview** button if pressed will still display the graph of the initial state, and the **Plot** button will display curves as fitted.
If you are happy with the result of the fit you may update the starting parameters by clicking two **From curr. fit** buttons:

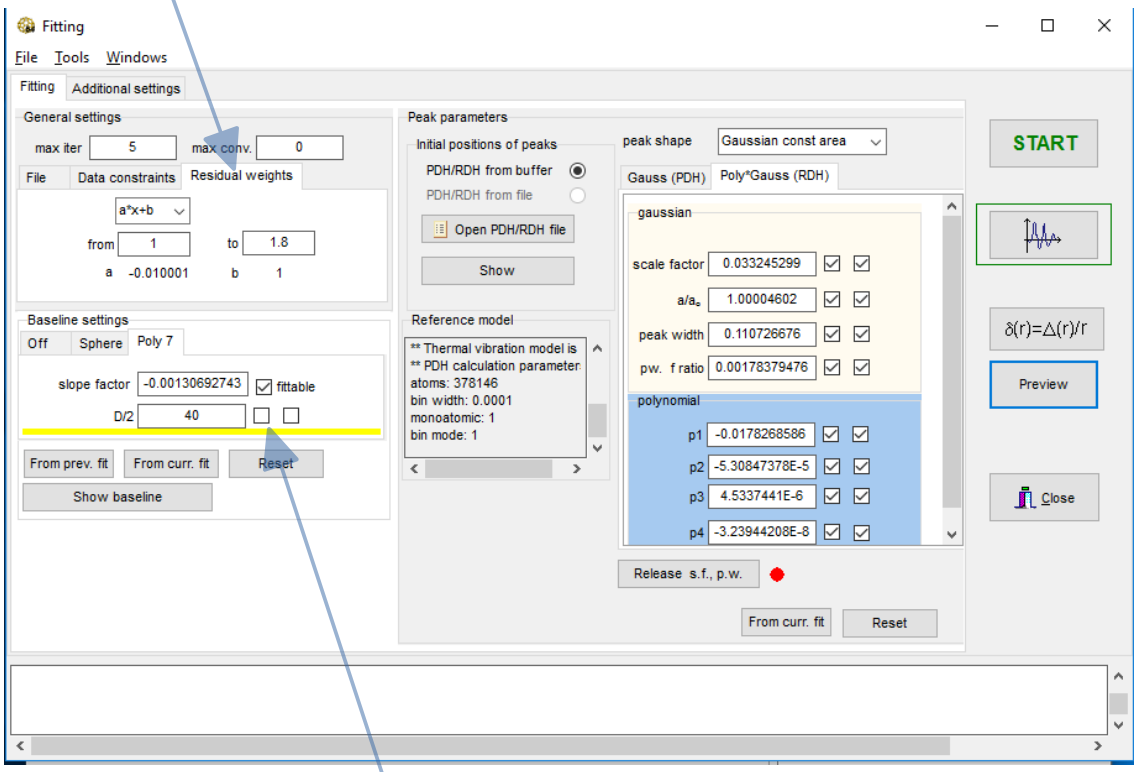



It is important to remember that our software does not update parameters automatically. Other fitting programs do, which may confuse persons who frequently do least squares. **Don't forget to update parameters after each successful run!** If you work with demo data do it now! The button on the left updates just the **slope factor**, the one on the right all the rest, including **D/2** which we don't fit at the moment.
The next step for the demo data is allowing the peak width to vary with distance in a linear manner. You do that by releasing the **pw.f.ratio**. Start fitting again. The match between the model and the data becomes better, but the shape of the baseline remains as that for a sphere. This is not the case for the demo data. Now you should release polynomial parameters **p1** to **p4** one by one, and each time do the fitting run and update parameters! If the result looks weird:

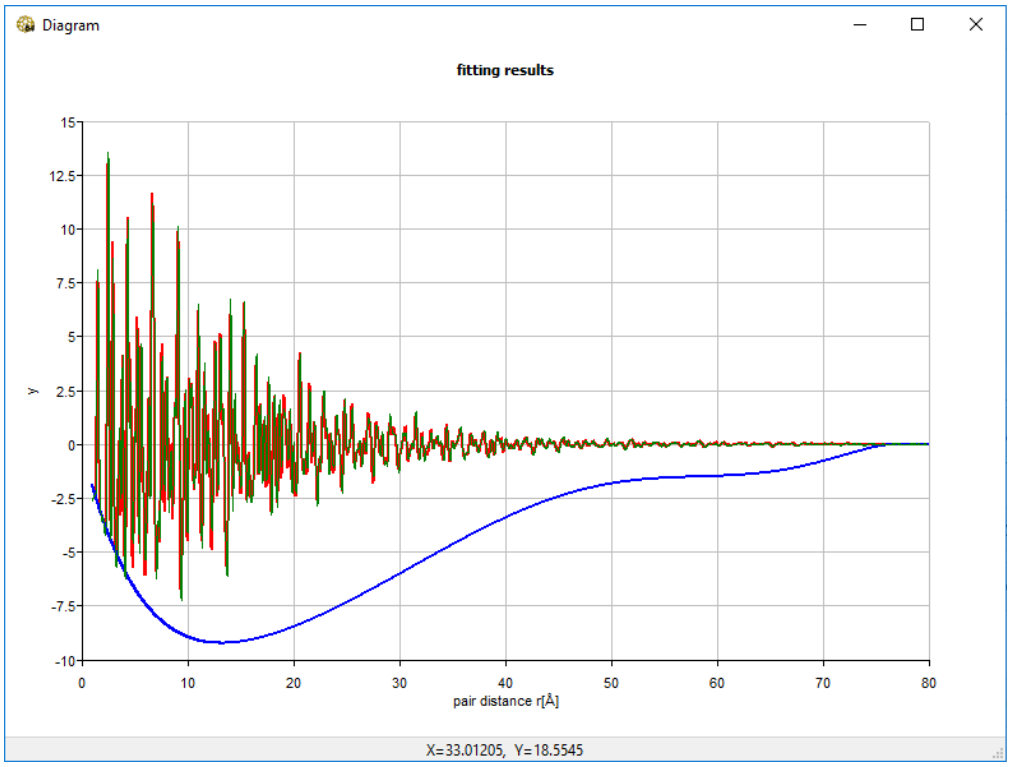



it's time to apply variable data weights.

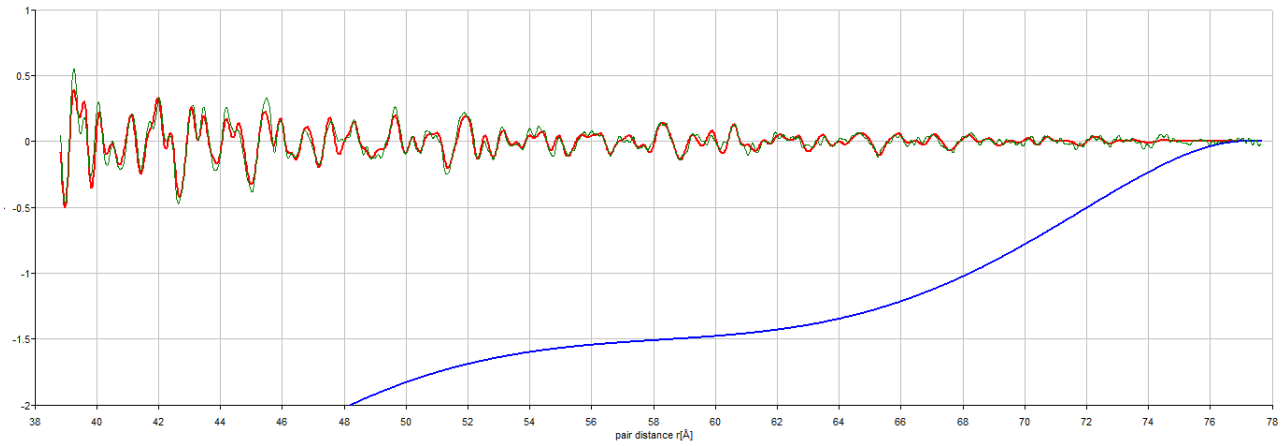
The “peaks” at the end of G(r) are quite small and the least squares procedure basically ignores them. **Residual weights** if activated change the data weights in a linear manner, in the demo case from 1 at the data beginning to 1.8 at the data end.



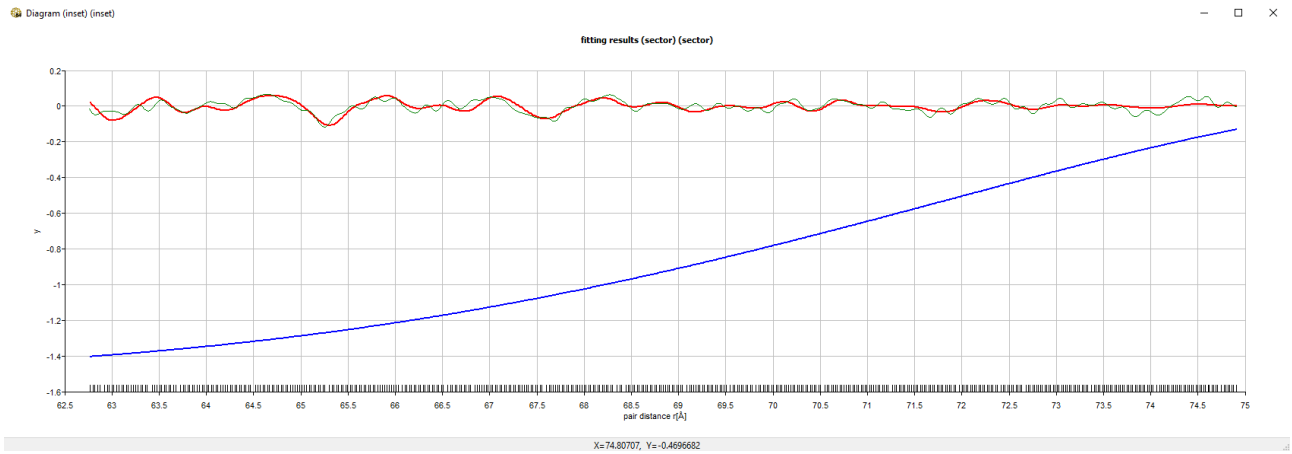

The values are to be chosen by trial-and-error method.
At the very last try to release **D/2** parameter.
The final fit should look like:



Zoom-in into the large distances part of the curve to check the fit quality.

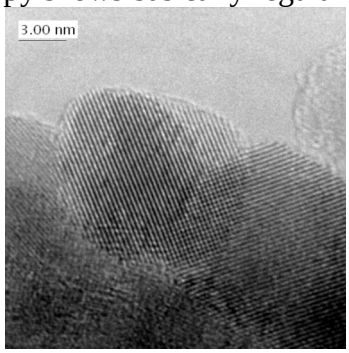



In our case the analytical G(r) is composed of 1976 heavily overlapping Gaussian peaks. You may set the **add peaks position to diagram** checkbox in the **Additional setting** tab to see where they are:



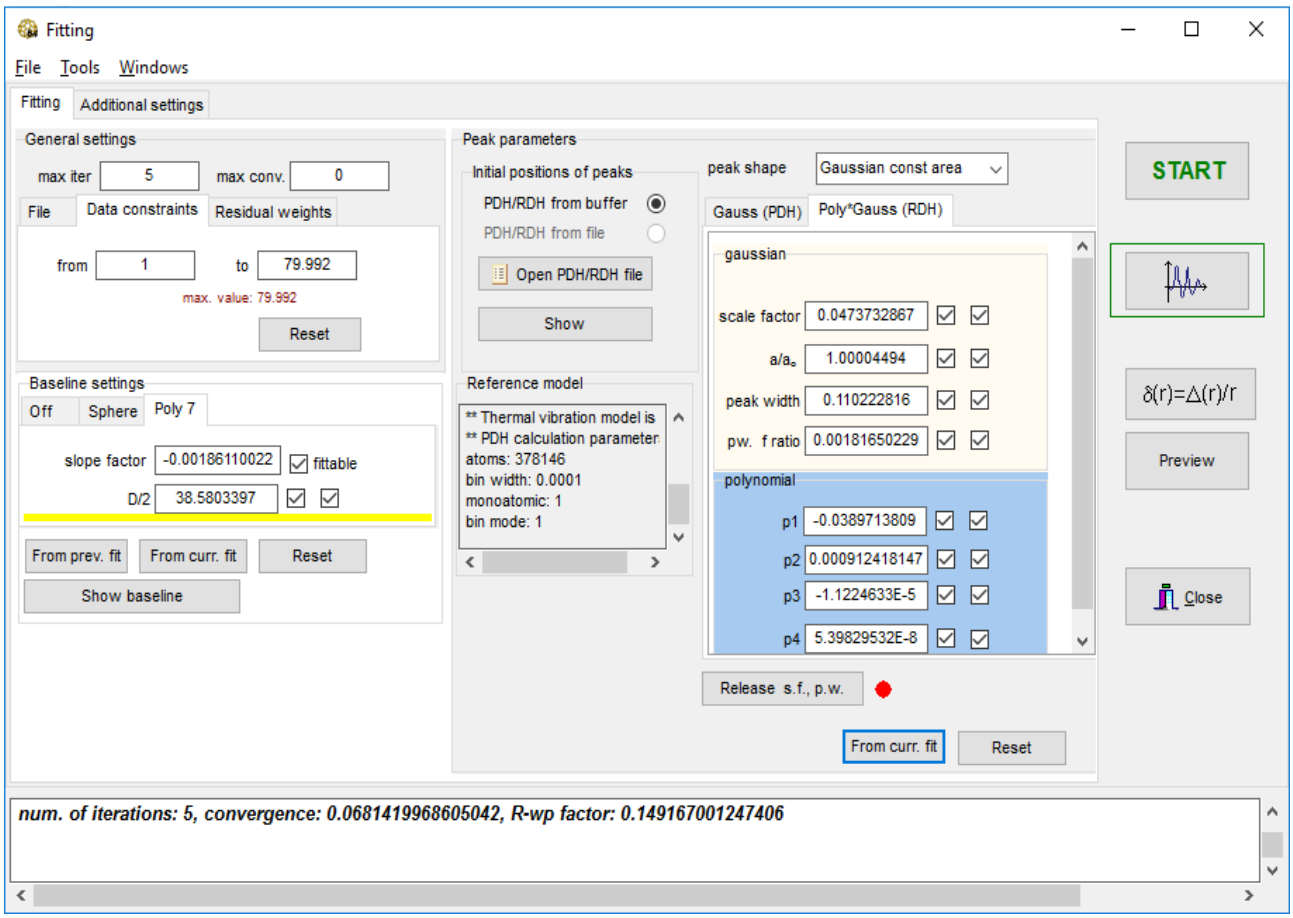

This shows that what appears to be peaks in G(r) are actually just data oscillations which are quite nicely reproduced by analytical G(r).

Note the shape of the baseline! It does not look like that for a sphere which we begun with, but rather suggests elongated shapes – plenty of long interatomic distances with similarly low abundance. However TEM microscopy shows basically regular grains:




The reason for such discrepancy is still unclear for us.

The final fit parameters are shown in the image below.


Fitting
File Tools Windows
Fitting
Additional settings
General settings
max iter 5
max conv. 0
File
Data constraints
Residual weights
from 1
to 79.992
max. value: 79.992
Reset
Baseline settings
Off
Sphere
Poly 7
slope factor -0.00186110022
fittable
D/2 38.5803397
From prev. fit
From curr. fit
Reset
Show baseline
Peak parameters
Initial positions of peaks
PDH/RDH from buffer
PDH/RDH from file
Open PDH/RDH file
Show
Reference model
** Thermal vibration model is
** PDH calculation parameter
atoms: 378146
bin width: 0.0001
monoatomic: 1
bin mode: 1
peak shape
Gaussian const area
Gauss (PDH)
Poly*Gauss (RDH)
gaussian
scale factor 0.0473732867
a/a0 1.00004494
peak width 0.110222816
pw. f ratio 0.00181650229
polynomial
p1 -0.0389713809
p2 0.000912418147
p3 -1.1224633E-5
p4 5.39829532E-8
Release s.f., p.w.
From curr. fit
Reset
START
δ(r)=Δ(r)/r
Preview
Close
num. of iterations: 5, convergence: 0.0681419968605042, R-wp factor: 0.149167001247406