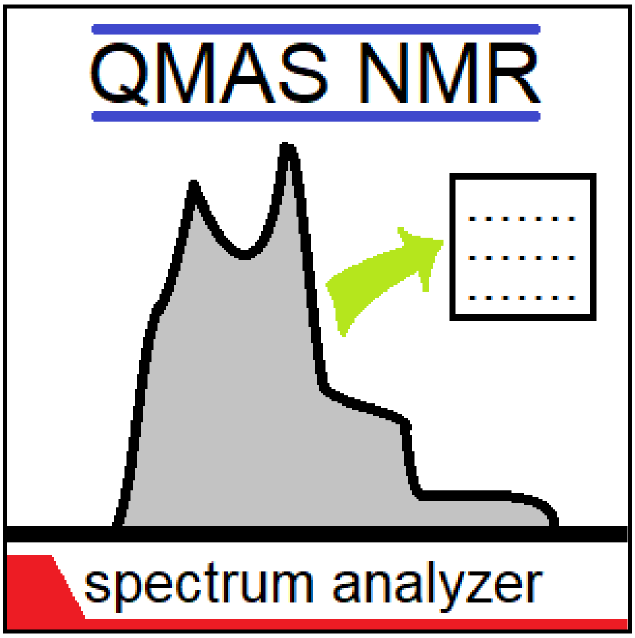
User's manual

To the program

QMAS NMR Spectrum Analyzer

Version 1.0



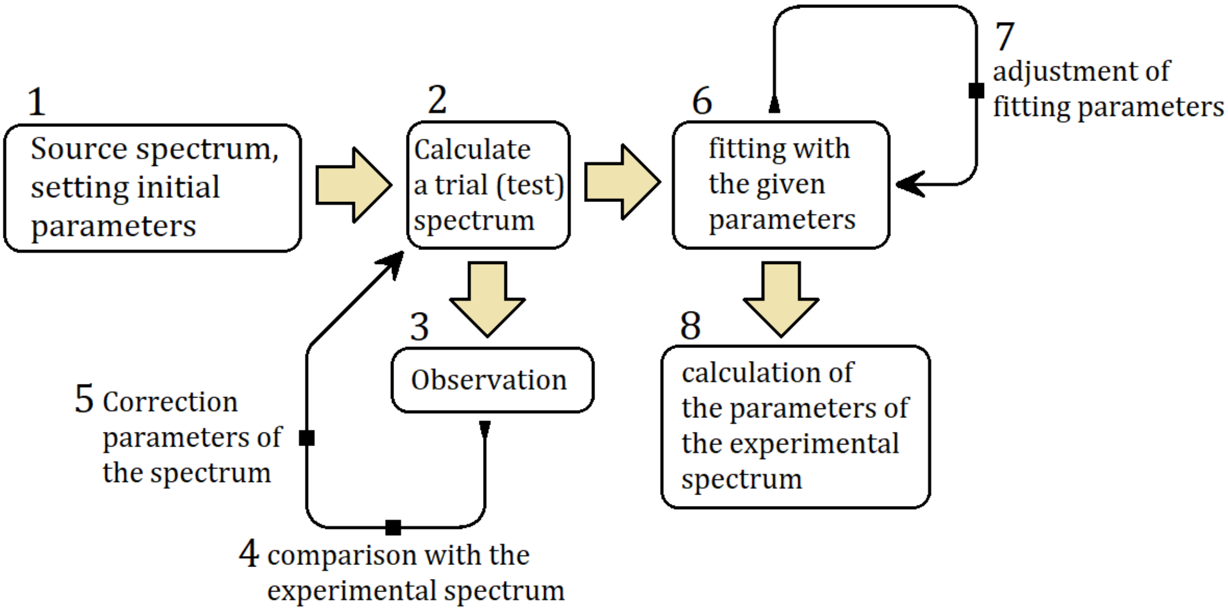
Feb 2020

Content

What is this program for?

This software package is designed to calculate and approximate the NMR spectrum of quadrupole nuclei in a polycrystalline solid during fast rotation of the sample at a magic angle for the center line transition (-1/2 - +1/2) in the second order of perturbation theory[1].

Schematic diagram of work with the program interface:


1
Source spectrum, setting initial parameters
2
Calculate a trial (test) spectrum
3
Observation
4 comparison with the experimental spectrum
5 Correction parameters of the spectrum
6
fitting with the given parameters
7
adjustment of fitting parameters
8
calculation of the parameters of the experimental spectrum

In this diagram, the arrows indicate all the possible paths that the user can go when working with the program. The longest one (and most complete) is the path corresponding to the sequential execution of all steps, marked with numbers 1-8. All steps are listed below:

1. To work with the program, you need to select the initial experimental spectrum, the parameters of which need to be clarified. Also the user has to set the initial parameters for the trial (test) spectrum, which can be adjusted by varying the parameters.

[1] For reference, in this case, the transition frequency is described as follows:

$$\omega^{(2)fastMAS}_{-\frac{1}{2},+\frac{1}{2}} = \omega_0 - \frac{1}{6\omega_0}\omega_Q{}^2\left\{I(I+1)-\frac{3}{4}\right\}\cdot\{D(\varphi,\eta)\cos^4\theta + E(\varphi,\eta)\cos^2\theta + F(\varphi,\eta)\},$$

where:

$$D(\varphi,\eta) = \frac{21}{16} - \frac{7}{8}\eta\cos(2\varphi) + \frac{7}{48}(\eta\cos(2\varphi))^2$$
$$E(\varphi,\eta) = -\frac{9}{8} + \frac{1}{12}\eta^2 + \eta\cos(2\varphi) - \frac{7}{24}(\eta\cos(2\varphi))^2$$
$$F(\varphi,\eta) = \frac{5}{16} - \frac{1}{8}\eta\cos(2\varphi) + \frac{7}{48}(\eta\cos(2\varphi))^2$$
$$\omega_Q = \frac{3e^2qQ}{2I(2I-1)\hbar}$$

2. A single spectrum calculation with the indicated parameters is performed. Its figure is being built.
3. The graphs of the experimental and calculated spectra are imposed on each other during construction.
4. So after the observations, you can notice their differences.
5. Based on the nature of the difference in the spectra, the user can adjust the calculation parameters of the probe spectrum to ensure convergence when fitting.
6. The calculated spectrum is adjusted to the experimental one by varying the parameters for given approximation parameters (errors, parameter steps, approximation method, etc.).
7. To speed up the approximation or increase its accuracy, it is possible to adjust its parameters.
8. After the approximation process is completed (the iteration stop condition is fulfilled or at the user's discretion), the parameters will immediately be available at which the best approximation for the experimental spectrum is achieved. These parameters are the result of the program processing.

Types of work schemes (indicating the numbers of steps that are performed):

А) Conventional manual way: **1** -> {**2** -> **3** -> **4** -> **5**}×n, n ≥ 1

Б) Express manual way: **1** -> {**2** -> **3** -> **4** -> **5**}×2 -> **6** -> **8**

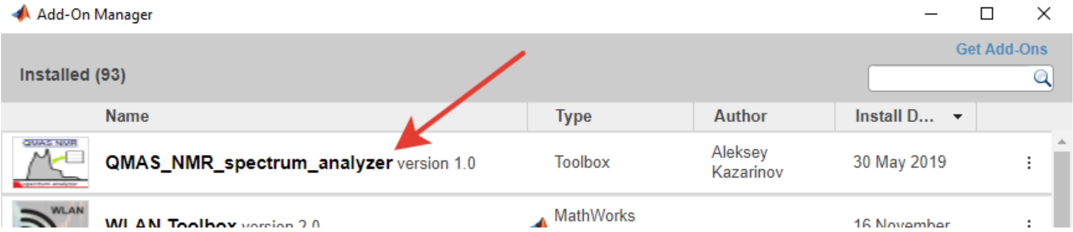
В) Accurate way: **1** -> {**2** -> **3** -> **4** -> **5**}×n -> {**6** -> **7**}×m -> **8,** n ≥ 2, m ≥ 2)

Package installation

The software package comes in the form of a MATLAB Toolbox with the name: QMAS_NMR_spectrum_analyzer.mltbx.

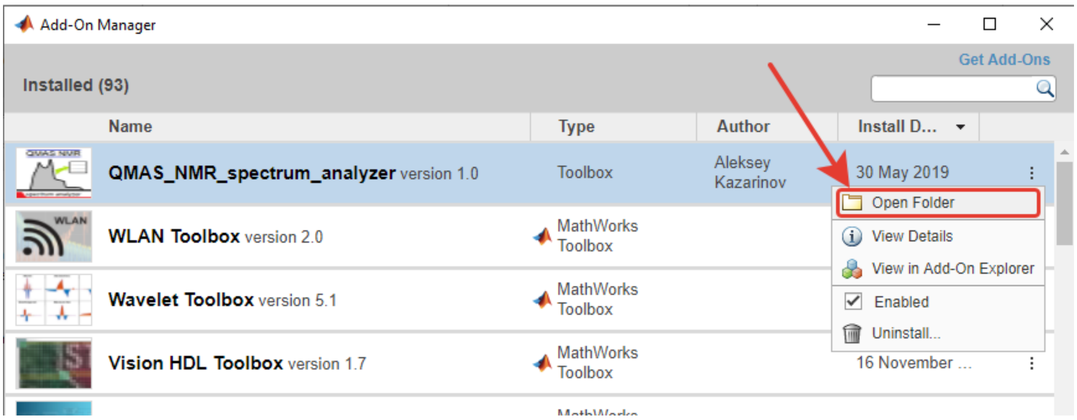
To install[2] this package on Windows, you have to:

1) Launch MATLAB via menu «**Start**».
2) Press the button «**Open**» -> «**Open…**».
3) In the explorer that opens, go to the location of the downloaded package and select it by clicking on the appropriate button. Then there is the opened window **Add-one Manager**, and automatic installation will start.
4) Upon completion of installation (Message «*Installation completed*») press the button «**Finish**». In the list of installed packages appears «QMAS_NMR_spectrum_analyzer»:



Name	Type	Author	Install D...
QMAS_NMR_spectrum_analyzer version 1.0	Toolbox	Aleksey Kazarinov	30 May 2019
WLAN Toolbox version 2.0	MathWorks		16 November

5) To go to the package directory, click on the "..." icon on the right, and select «**Open Folder**»:



Name	Type	Author	Install D...
QMAS_NMR_spectrum_analyzer version 1.0	Toolbox	Aleksey Kazarinov	30 May 2019
WLAN Toolbox version 2.0	MathWorks Toolbox		
Wavelet Toolbox version 5.1	MathWorks Toolbox		
Vision HDL Toolbox version 1.7	MathWorks Toolbox		16 November ...

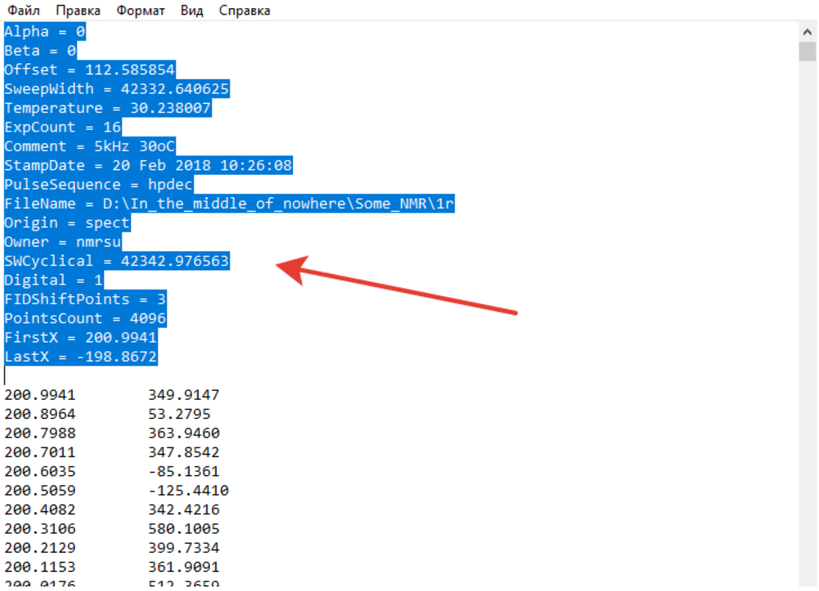
Installation completed!

[2] This method was tested in MATLAB version 2018b, so there may be slight differences for other versions. The essence of the installation method does not change from this. The package is compatible with version 2014a and later.

Beginning of work

To start processing the spectra, firstly you have to prepare a file with the spectrum. The program is able to process files with the «***.txt**» extension without header.

If the file has a header, for example:

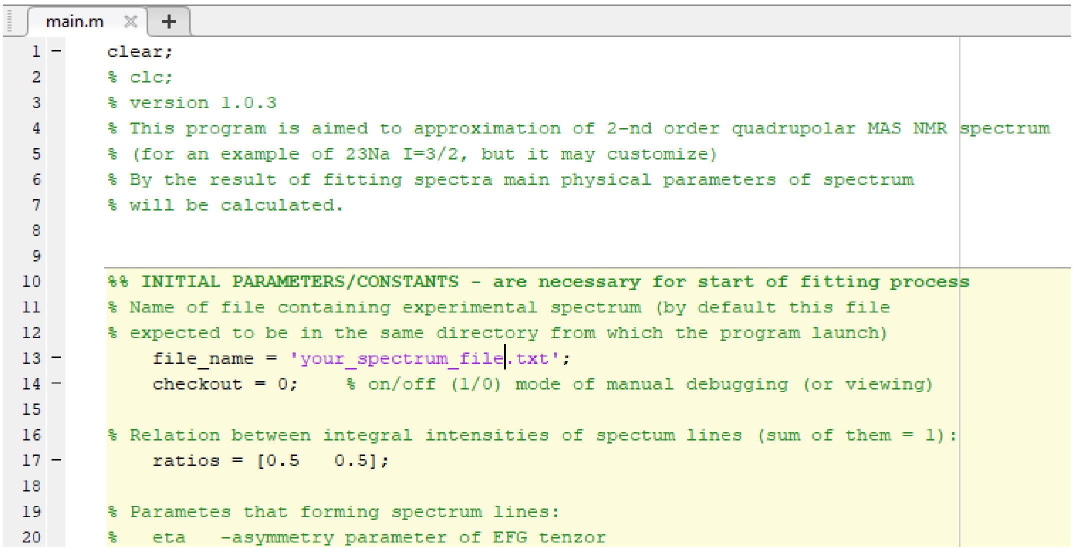


```
Alpha = 0
Beta = 0
Offset = 112.585854
SweepWidth = 42332.640625
Temperature = 30.238007
ExpCount = 16
Comment = 5kHz 30oC
StampDate = 20 Feb 2018 10:26:08
PulseSequence = hpdec
FileName = D:\In_the_middle_of_nowhere\Some_NMR\1r
Origin = spect
Owner = nmrsu
SWCyclical = 42342.976563
Digital = 1
FIDShiftPoints = 3
PointsCount = 4096
FirstX = 200.9941
LastX = -198.8672

200.9941        349.9147
200.8964        53.2795
200.7988        363.9460
200.7011        347.8542
200.6035        -85.1361
200.5059        -125.4410
200.4082        342.4216
200.3106        580.1005
200.2129        399.7334
200.1153        361.9091
200.0176        512.3659
```

Then it must be deleted so that the numerical values go from the very beginning (without empty lines in front of them).

The prepared file should be placed in the same directory with the installed package.

Next you need to open the file «**main.m**» in MATLAB. On the beginning of the file, at the section of «**Initial data/constants**» you should to set up a value of variable **file_name**, by input of file name of a spectrum to be processed, as shown below:



```
clear;
% clc;
% version 1.0.3
% This program is aimed to approximation of 2-nd order quadrupolar MAS NMR spectrum
% (for an example of 23Na I=3/2, but it may customize)
% By the result of fitting spectra main physical parameters of spectrum
% will be calculated.


%% INITIAL PARAMETERS/CONSTANTS - are necessary for start of fitting process
% Name of file containing experimental spectrum (by default this file
% expected to be in the same directory from which the program launch)
    file_name = 'your_spectrum_file.txt';
    checkout = 0;     % on/off (1/0) mode of manual debugging (or viewing)

% Relation between integral intensities of spectum lines (sum of them = 1):
    ratios = [0.5   0.5];

% Parametes that forming spectrum lines:
%   eta   -asymmetry parameter of EFG tenzor
```

Spectrum calculation

The program allows to perform calculation and to build a spectrum by input parameter. Preliminary spectrum calculation is needed for setting up the optimal parameters to start fitting.

To calculate a trial (test) spectrum, it is need to come to «**mode of manual debugging**». For this user has to set a value to the ***checkout***=1. Further, have to input parameters of lines (components consisting a whole spectrum) in file «**main.m**»:

1) Ratios between integral intensities of lines (components) that are weights. Several lines of spectrum can take place because of existing inequivalent positions of quadrupolar nuclei in a crystal to be investigated. n this case, the sum of the weights of the intensities of all lines is normalized to one: $\sum_i r_i = 1$, where r_i – weight of an line (component).
2) Asymmetry parameter of EFG - η.
   Here and below the parameters are specified for each line (component).
3) Quadrupole coupling constant $Chi = \frac{e^2qQ}{\hbar}$.
4) Isotropic chemical shift δ_{iso}. Formally, it is presented as a frequency shift relative to the median of the spectrum ("gravity center" in English literature). The parameters of the Lorentz and Gaussian lines, as well as the ratio α between their integral intensities.

Explanations of how to do this are given on the example of 2 lines (contributions) below.

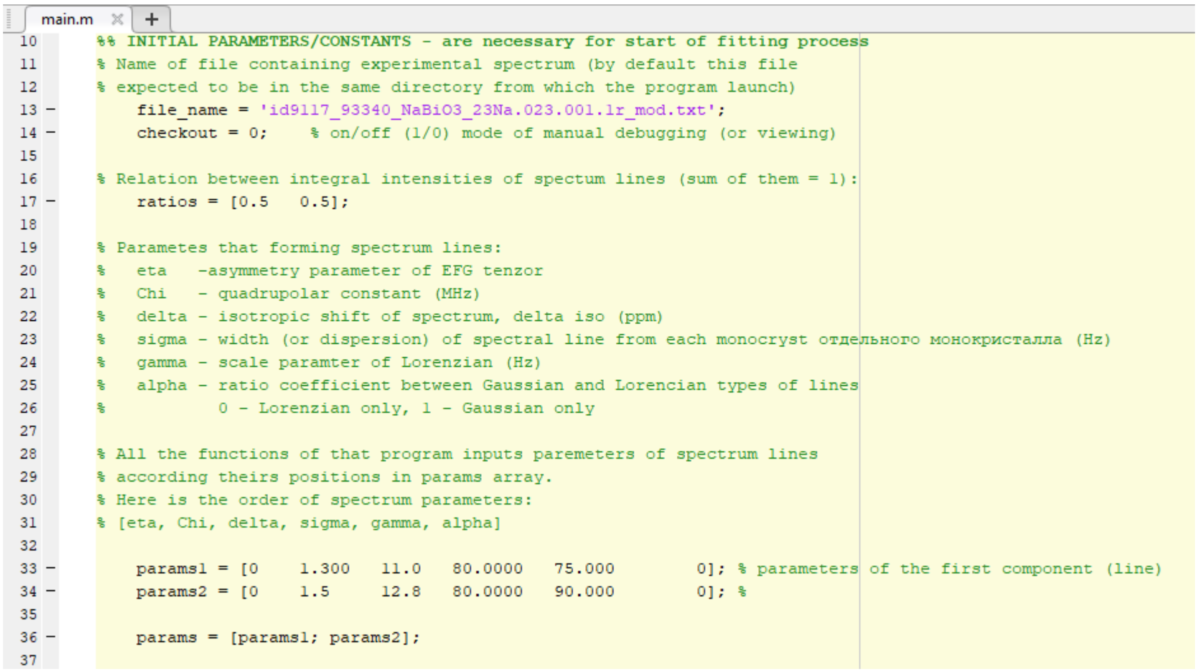
1) The relative integral intensities of the contributions (lines):
```
ratios = [0.384 0.616];
```
2) Then for each line its own parameters are set according to their position in the list [eta, Chi, delta, sigma, alpha], where:

eta	EFG asymmetry parameter
Chi	quadrupole constant (MHz)
delta	spectrum frequency shift, delta iso (ppm)
sigma	width (dispersion) of the spectral line from an individual single crystal (Hz)
gamma	Lorentz line scale parameter (Hz)
alpha	correlation between Gaussian and Lorentz types of lines (0 - only Lorenz, 1 - only Gaussian)

```
params1 = [0     1.2590    12.0    80.0000    78.000          0];
params2 = [0     1.511     13.0    80.0000    88.000          0];
```
3) Then these parameters should be packed into the **params** array of all spectrum parameters:
```
params = [params1; params2];
```

<u>Note!</u> If the spectrum consists of one line (a contribution), then it is necessary to set `params = [params1];`

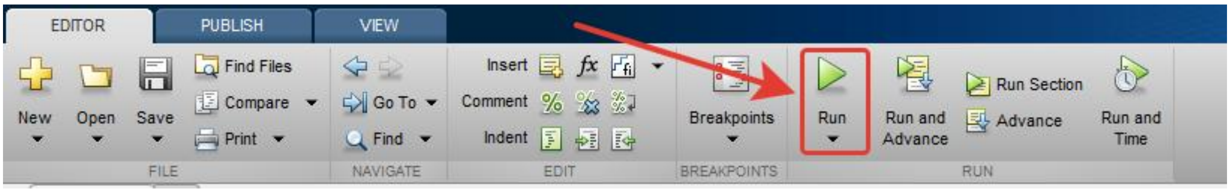
An example of the prepared initial parameters is presented below:



```
%% INITIAL PARAMETERS/CONSTANTS - are necessary for start of fitting process
% Name of file containing experimental spectrum (by default this file
% expected to be in the same directory from which the program launch)
    file_name = 'id9117_93340_NaBiO3_23Na.023.001.1r_mod.txt';
    checkout = 0;     % on/off (1/0) mode of manual debugging (or viewing)

% Relation between integral intensities of spectum lines (sum of them = 1):
    ratios = [0.5   0.5];

% Parametes that forming spectrum lines:
%   eta   -asymmetry parameter of EFG tenzor
%   Chi   - quadrupolar constant (MHz)
%   delta - isotropic shift of spectrum, delta iso (ppm)
%   sigma - width (or dispersion) of spectral line from each monocryst отдельного монокристалла (Hz)
%   gamma - scale paramter of Lorenzian (Hz)
%   alpha - ratio coefficient between Gaussian and Lorencian types of lines
%           0 - Lorenzian only, 1 - Gaussian only

% All the functions of that program inputs paremeters of spectrum lines
% according theirs positions in params array.
% Here is the order of spectrum parameters:
% [eta, Chi, delta, sigma, gamma, alpha]

    params1 = [0    1.300   11.0   80.0000   75.000        0]; % parameters of the first component (line)
    params2 = [0    1.5     12.8   80.0000   90.000        0]; %

    params = [params1; params2];
```

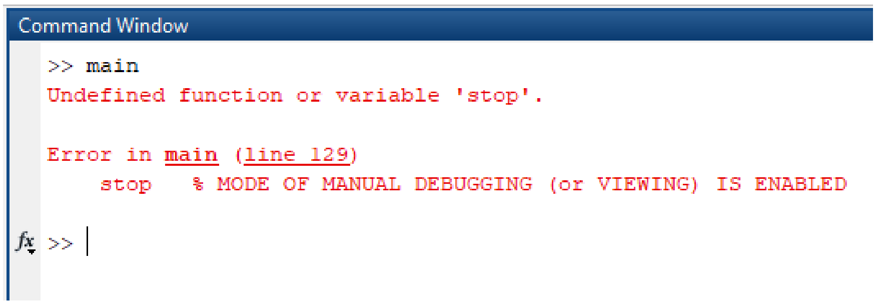
Next, you need to set the technical parameters (see the section «**Technical constants**» from the chapter «**Settings**» of this manual).

Now you can start the spectrum calculation by clicking on «**Run**» MATLAB menu:



The calculation will take some time (usually a few seconds), after which a characteristic sound signal will sound and a window with a graph will appear.

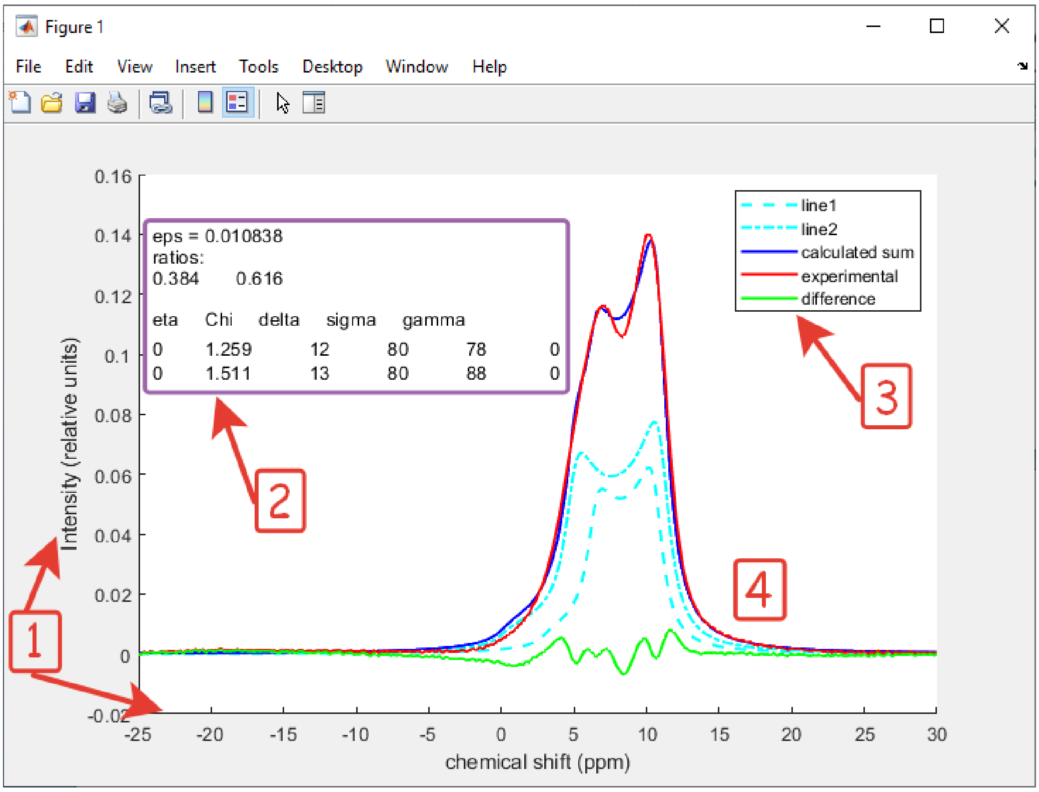
An exception will be thrown to the console caused by the debugging mode enabled, which was set earlier in the form of building a trial spectrum:



```
>> main
Undefined function or variable 'stop'.

Error in main (line 129)
    stop   % MODE OF MANUAL DEBUGGING (or VIEWING) IS ENABLED

>> |
```

It is OK. This message can be ignored. It signals that the calculation of the spectrum and the construction of its graph are completed.

Below is an example of a plot. Let's consider it in more detail.



```
eps = 0.010838
ratios:
0.384      0.616

eta    Chi    delta    sigma    gamma
0      1.259      12       80       78      0
0      1.511      13       80       88      0
```

Below are explanations for the figure:

1) The intensities of the experimental (red) and calculated spectra (blue) are integrally normalized to unity. Important remark: the axis of the chemical shift has a non-standard direction («-» -> «+»).
2) For convenience, a table is displayed showing the discrepancy between the calculated and experimental spectra (***eps=0.010838***) and the parameters of the calculated spectrum at which it is achieved (set earlier). The discrepancy (error) is calculated according to the selected method (default: standard deviation with weights proportional to the intensity of the experimental spectrum).
3) In the legend, the lines are displayed in the order in which they follow in the given ratios and **params** arrays (see earlier). The first two lines are displayed with a dash and dash-dot lines, respectively. The remaining lines (if any) are indicated by dots. Difference between the experimental and calculated spectra is marked in green color.
4) Focusing on the differences between the spectra, it is possible to set more suitable parameters for the contributions, which are displayed separately for

convenience. More subtle manual adjustment is possible when taking into account the green graph showing the discrepancy.

Spectrum approximation

Before proceeding to the fitting of the spectrum, you need to be sure that the initial parameters are set correctly, namely:

A) Suitable number of lines (deposits) is specified,

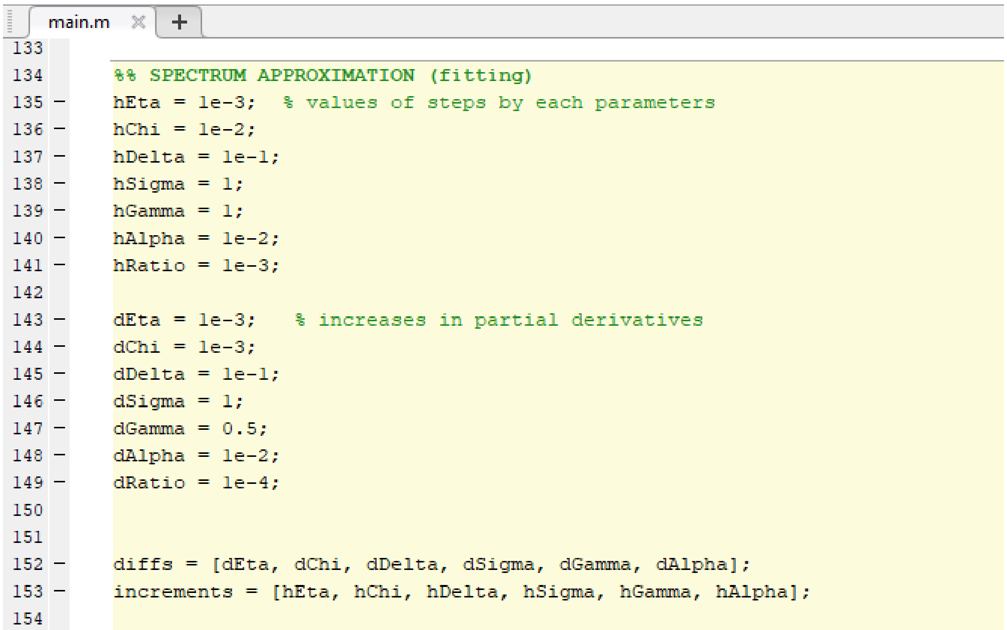
B) Their acceptable relative weights are set.

This criterion requires some experience to satisfy it. It is often not so difficult. You need to try several ways of arrangement of lines and select the appropriate way.

The spectrum fitting is performed via the gradient descent method, and the minimum of the error function (discrepancy between the spectra) is searched.

Parameters setting

On section «**Spectrum approximation**» in **main.m** file, it is possible to set values of steps for each contribution parameter and increases for partial derivatives of each parameter in gradient descent method:



```
%% SPECTRUM APPROXIMATION (fitting)
hEta = 1e-3;  % values of steps by each parameters
hChi = 1e-2;
hDelta = 1e-1;
hSigma = 1;
hGamma = 1;
hAlpha = 1e-2;
hRatio = 1e-3;

dEta = 1e-3;   % increases in partial derivatives
dChi = 1e-3;
dDelta = 1e-1;
dSigma = 1;
dGamma = 0.5;
dAlpha = 1e-2;
dRatio = 1e-4;


diffs = [dEta, dChi, dDelta, dSigma, dGamma, dAlpha];
increments = [hEta, hChi, hDelta, hSigma, hGamma, hAlpha];

```

The general rule for setting these values is: $dValue \ll hValue$

Herewith, $\boldsymbol{dValue}$ should not be too small so that the derivative by $Value$ does not take such small values that they would be comparable with the accuracy of calculations in MATLAB.

The upper limit on the value of $\boldsymbol{hValue}$ is determined taking into account the sensitivity of the spectral lines to the corresponding parameter.

Experience shows that the magnitude of the sensitivity of the spectrum to the parameters (from highest to lowest) is as follows:

- **Ratio** (hRatio)
- **Chi** (hChi)
- **Delta** (…)
- **Eta**
- **Sigma/Gamma/alpha**

Therefore, it is recommended that you select $\boldsymbol{hValue}$ at the top of the list significantly less (~10^{-3}), than at the bottom ones (~10^{0}). Arrays ***diffs*** and ***increments*** are not allowed to adjust in order to avoid problems with the program.

The spectrum is approximated by iterations. At each iteration, increments are performed over all line parameters allowed for variation. Iterations stop when the maximum number of iterations specified in ***max_iter*** is reached, or acceptable approximation error is reached. The acceptable value is customized via ***epsilon*** parameter.

User can specify parameters which ones to adjust. This opportunity is realized through comment/uncomment "**stepLineParameter(..)**" fitting functions corresponding each contribution. The general view of string for adjustment of any parameter (besides intensities) is as follows:

```
[params,eps,G_w]= stepLineParameter(n_line,n_param,dValue,hValue,…,params,…);
```

Here:

stepLineParameter – function to performing gradient descent by a specified parameter.

[params, eps, G_w] – respective descriptions to output values:

- A list of all the paraters of general spectrum (taking into account a new value of adjusted parameter)
- Error value after change by using new value of parameter by which gradient descent have performed
- New calculated general spectrum

n_line – number of a line (contribution) of the general spectrum (number of parameters set in **params** array of all the parameters of general spectrum). The parameter of the line is being adjusted.

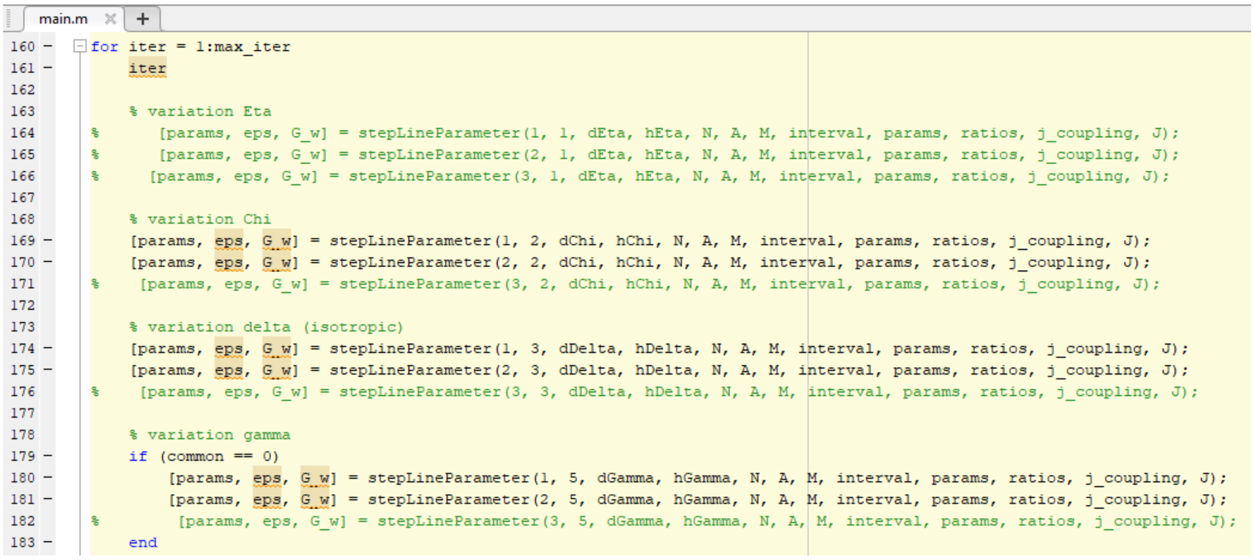
n_param – an order number of parameter in the list of parameters for the selected line (through **n_line** argument) that points on one from the following array: [eta, Chi, delta, sigma, alpha]. I.e. **eta** -> **1**, **Chi** -> **2**, **delta** -> **3** and so on.

dValue, ***hValue*** – values of increase and step by ***Value*** parameter (Chi, eta, …)

For instance, you want to switch on fitting by ***delta*** parameter of second line (contribution), then you have to write the following string:

```
[params,eps,G_w]= stepLineParameter(2,3,dDelta,hDelta,...,params,...);
```

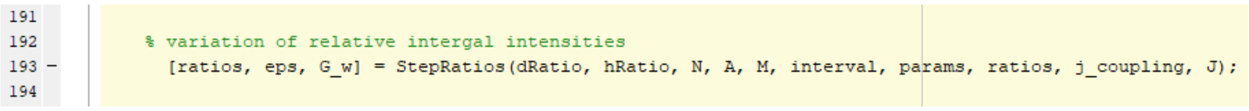
An example of setting up fitting of parameters for 2 lines is presented on a figure below:



```
for iter = 1:max_iter
    iter

    % variation Eta
%      [params, eps, G_w] = stepLineParameter(1, 1, dEta, hEta, N, A, M, interval, params, ratios, j_coupling, J);
%      [params, eps, G_w] = stepLineParameter(2, 1, dEta, hEta, N, A, M, interval, params, ratios, j_coupling, J);
%     [params, eps, G_w] = stepLineParameter(3, 1, dEta, hEta, N, A, M, interval, params, ratios, j_coupling, J);

    % variation Chi
    [params, eps, G_w] = stepLineParameter(1, 2, dChi, hChi, N, A, M, interval, params, ratios, j_coupling, J);
    [params, eps, G_w] = stepLineParameter(2, 2, dChi, hChi, N, A, M, interval, params, ratios, j_coupling, J);
%    [params, eps, G_w] = stepLineParameter(3, 2, dChi, hChi, N, A, M, interval, params, ratios, j_coupling, J);

    % variation delta (isotropic)
    [params, eps, G_w] = stepLineParameter(1, 3, dDelta, hDelta, N, A, M, interval, params, ratios, j_coupling, J);
    [params, eps, G_w] = stepLineParameter(2, 3, dDelta, hDelta, N, A, M, interval, params, ratios, j_coupling, J);
%    [params, eps, G_w] = stepLineParameter(3, 3, dDelta, hDelta, N, A, M, interval, params, ratios, j_coupling, J);

    % variation gamma
    if (common == 0)
        [params, eps, G_w] = stepLineParameter(1, 5, dGamma, hGamma, N, A, M, interval, params, ratios, j_coupling, J);
        [params, eps, G_w] = stepLineParameter(2, 5, dGamma, hGamma, N, A, M, interval, params, ratios, j_coupling, J);
%        [params, eps, G_w] = stepLineParameter(3, 5, dGamma, hGamma, N, A, M, interval, params, ratios, j_coupling, J);
    end
```

It shows that fitting by ***eta*** is off, and ***Chi***, ***delta*** и ***gamma*** are set for adjustment for both lines. Also variation of ***gamma*** happens under switched off shared line parameters (***common***=*0*), i.e. increase of **gamma** parameter is produced separately for each line.

A variation of relative integral intensities does not require any special (additional) settings from your side. If you wish the variation can be switched off. Then relative weights of lines will be constant.

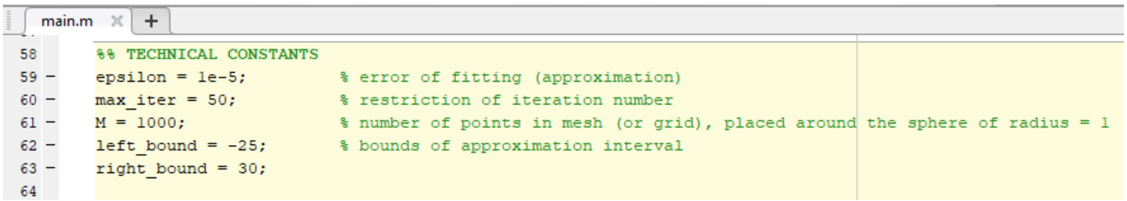


```
    % variation of relative intergal intensities
      [ratios, eps, G_w] = StepRatios(dRatio, hRatio, N, A, M, interval, params, ratios, j_coupling, J);
```

Approximation

A process of approximation can be launched under the following conditions:

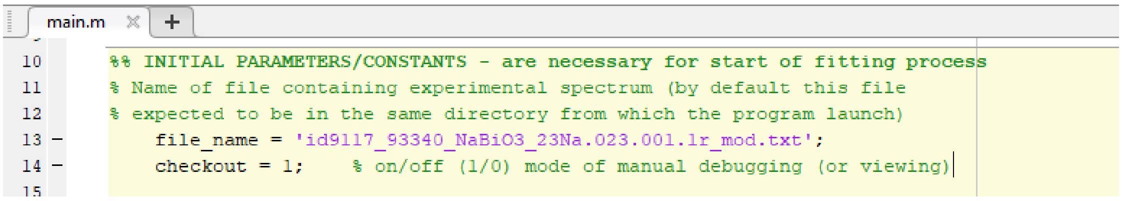
- Earlier have been calculated the trial (probe) spectrum (see the previous section).
- In section «**Technical constants**» are written values for ***epsilon***, ***max_iter***, ***left_bound***, ***right_bound***, as shown follow:



```
%% TECHNICAL CONSTANTS
epsilon = 1e-5;          % error of fitting (approximation)
max_iter = 50;           % restriction of iteration number
M = 1000;                % number of points in mesh (or grid), placed around the sphere of radius = 1
left_bound = -25;        % bounds of approximation interval
right_bound = 30;
```

left_bound, ***right_bound*** (in ppm) form an interval of chemical shift axis, within which approximation/fitting is performed.

- In section «**Initial parameters/constants**» ***checkout*** flag is set to 0:

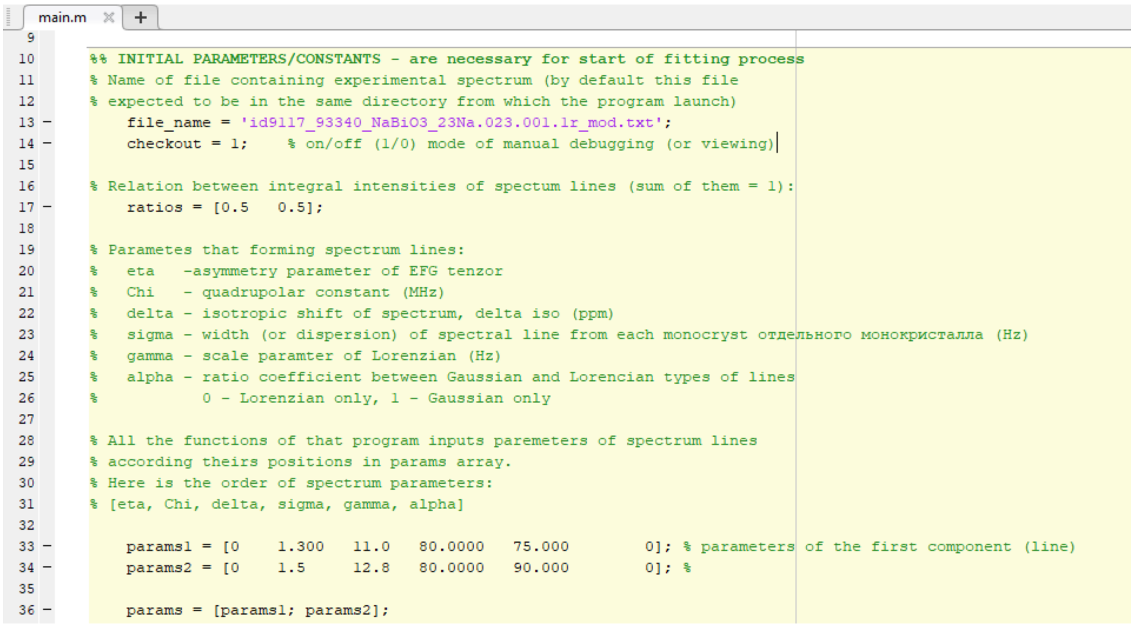


```
%% INITIAL PARAMETERS/CONSTANTS - are necessary for start of fitting process
% Name of file containing experimental spectrum (by default this file
% expected to be in the same directory from which the program launch)
    file_name = 'id9117_93340_NaBiO3_23Na.023.001.1r_mod.txt';
    checkout = 1;     % on/off (1/0) mode of manual debugging (or viewing)
```

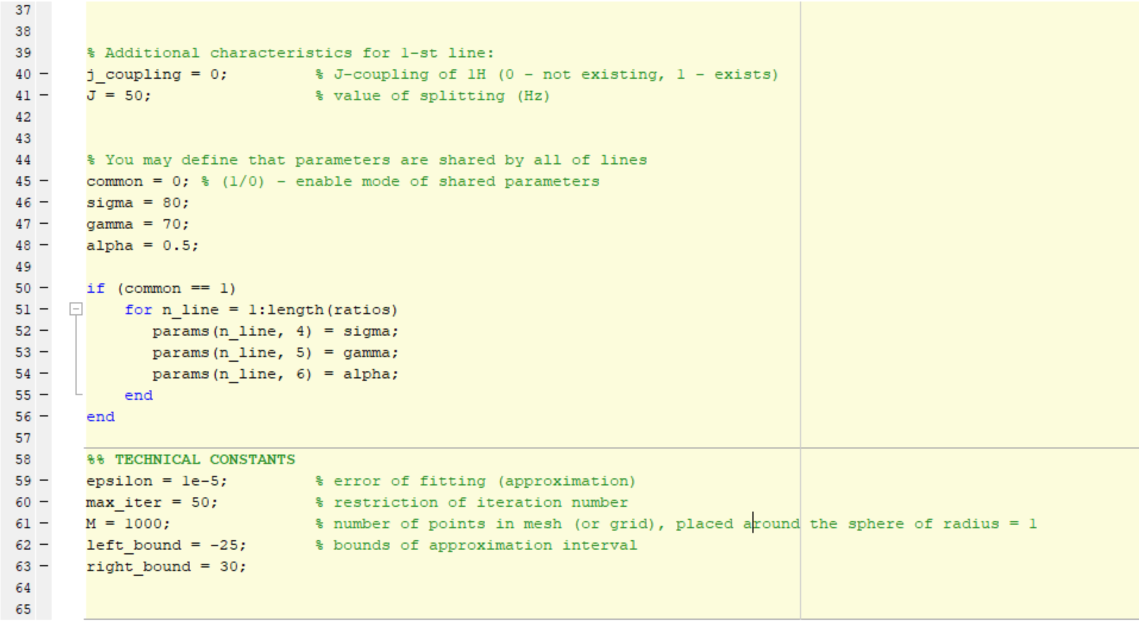
- Desired values of steps by each parameter and increments are set (***hValue***, ***dValue***) (see earlier).
- Necessary iterations of parameters are written. (see earlier)

Then for launch of fitting/approximation you can press the button «**Run**».

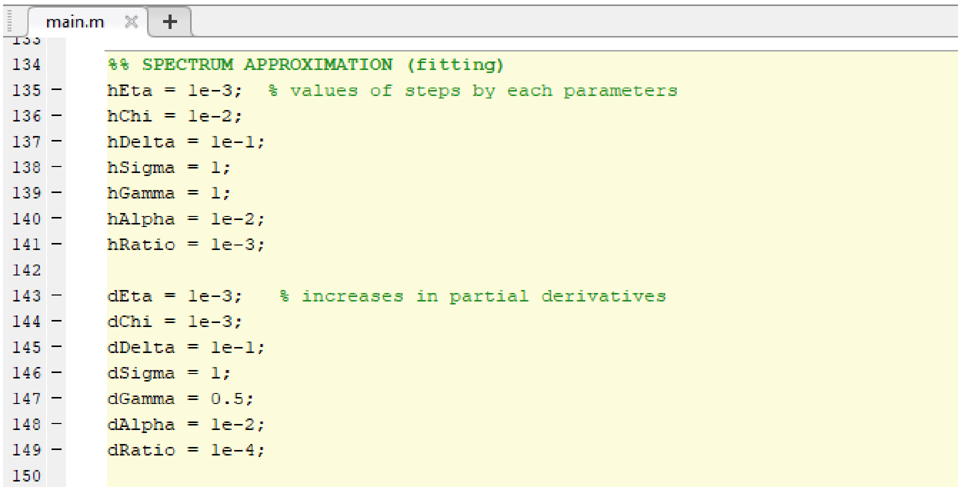
<u>Example:</u> Approximation of spectrum consisting of 2 lines (contributions). Selected settings of fitting are shown below:



```

%% INITIAL PARAMETERS/CONSTANTS - are necessary for start of fitting process
% Name of file containing experimental spectrum (by default this file
% expected to be in the same directory from which the program launch)
    file_name = 'id9117_93340_NaBiO3_23Na.023.001.1r_mod.txt';
    checkout = 1;     % on/off (1/0) mode of manual debugging (or viewing)

% Relation between integral intensities of spectum lines (sum of them = 1):
    ratios = [0.5   0.5];

% Parametes that forming spectrum lines:
%   eta   -asymmetry parameter of EFG tenzor
%   Chi   - quadrupolar constant (MHz)
%   delta - isotropic shift of spectrum, delta iso (ppm)
%   sigma - width (or dispersion) of spectral line from each monocryst отдельного монокристалла (Hz)
%   gamma - scale paramter of Lorenzian (Hz)
%   alpha - ratio coefficient between Gaussian and Lorencian types of lines
%           0 - Lorenzian only, 1 - Gaussian only

% All the functions of that program inputs paremeters of spectrum lines
% according theirs positions in params array.
% Here is the order of spectrum parameters:
% [eta, Chi, delta, sigma, gamma, alpha]

    params1 = [0    1.300   11.0   80.0000   75.000        0]; % parameters of the first component (line)
    params2 = [0    1.5     12.8   80.0000   90.000        0]; %

    params = [params1; params2];
```

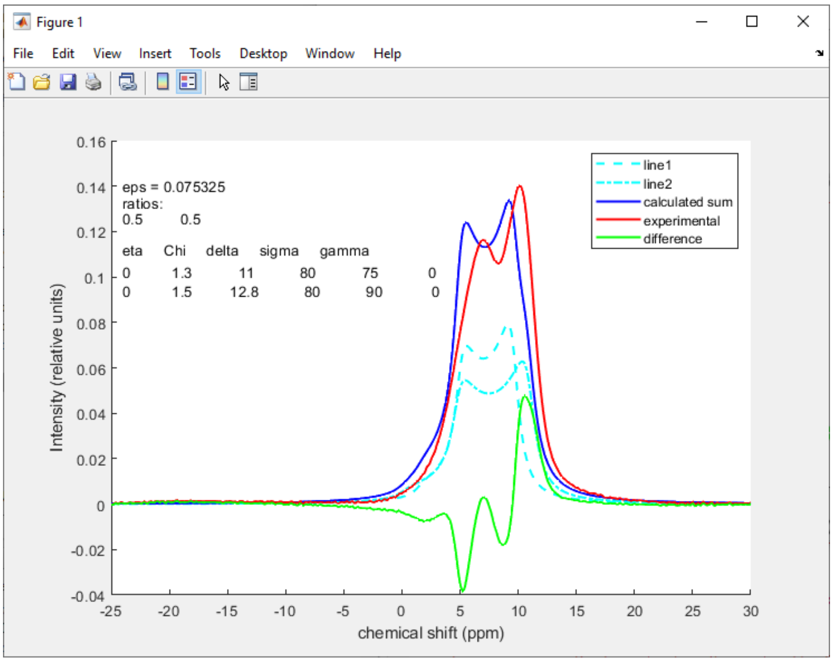


```
% Additional characteristics for 1-st line:
j_coupling = 0;          % J-coupling of 1H (0 - not existing, 1 - exists)
J = 50;                  % value of splitting (Hz)


% You may define that parameters are shared by all of lines
common = 0; % (1/0) - enable mode of shared parameters
sigma = 80;
gamma = 70;
alpha = 0.5;

if (common == 1)
    for n_line = 1:length(ratios)
       params(n_line, 4) = sigma;
       params(n_line, 5) = gamma;
       params(n_line, 6) = alpha;
    end
end

%% TECHNICAL CONSTANTS
epsilon = 1e-5;          % error of fitting (approximation)
max_iter = 50;           % restriction of iteration number
M = 1000;                % number of points in mesh (or grid), placed around the sphere of radius = 1
left_bound = -25;        % bounds of approximation interval
right_bound = 30;
```



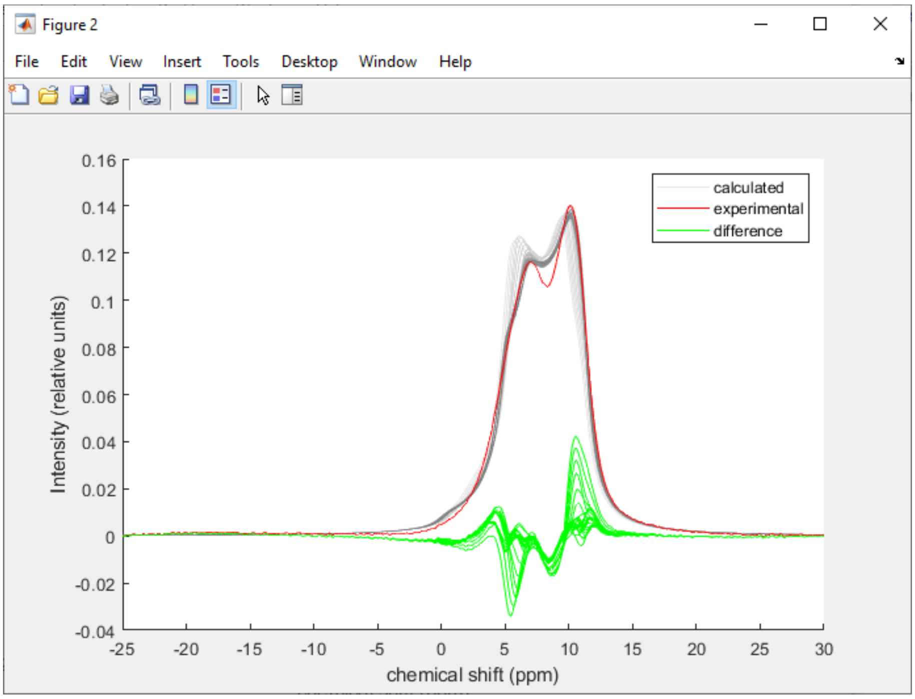
```
%% SPECTRUM APPROXIMATION (fitting)
hEta = 1e-3;  % values of steps by each parameters
hChi = 1e-2;
hDelta = 1e-1;
hSigma = 1;
hGamma = 1;
hAlpha = 1e-2;
hRatio = 1e-3;

dEta = 1e-3;   % increases in partial derivatives
dChi = 1e-3;
dDelta = 1e-1;
dSigma = 1;
dGamma = 0.5;
dAlpha = 1e-2;
dRatio = 1e-4;
```

If stop the fitting (combination **Ctrl^C**) after 23 iterations, the program will plot 2 figures and write a final message into console.

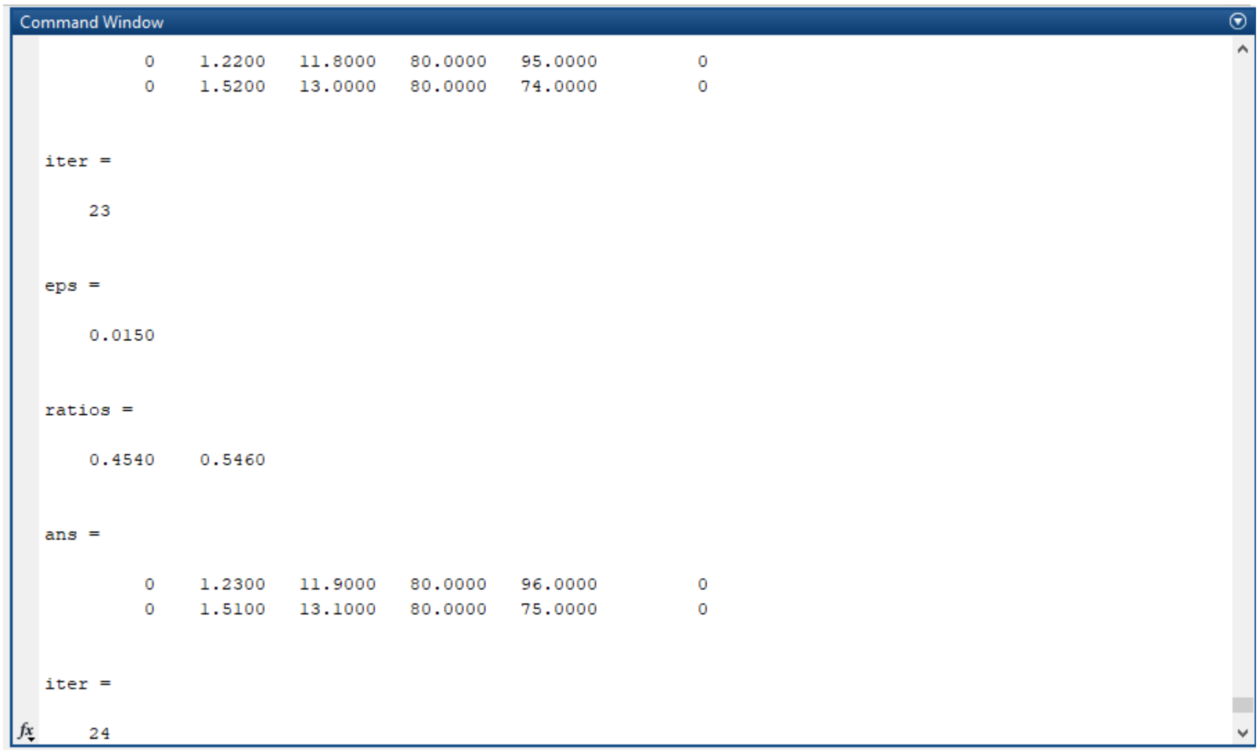


```
eps = 0.075325
ratios:
0.5      0.5

eta    Chi    delta    sigma    gamma
0      1.3     11       80       75        0
0      1.5     12.8     80       90        0
```

On the first figure (upper one) is shown the source spectrum corresponding to initial parameters. On the second one (that is lower) it is shown how the spectrum was being changed step by step. The more ordinal number of the step is the respective spectrum is darker. This figure gives a reason to estimate a grade of success for approximation.



A message in the console contains parameters of the contributions (lines) which have been reached as a result of approximation process on each step of iteration.



```
         0    1.2200   11.8000   80.0000   95.0000         0
         0    1.5200   13.0000   80.0000   74.0000         0


iter =

    23


eps =

    0.0150


ratios =

    0.4540    0.5460


ans =

         0    1.2300   11.9000   80.0000   96.0000         0
         0    1.5100   13.1000   80.0000   75.0000         0


iter =

    24
```

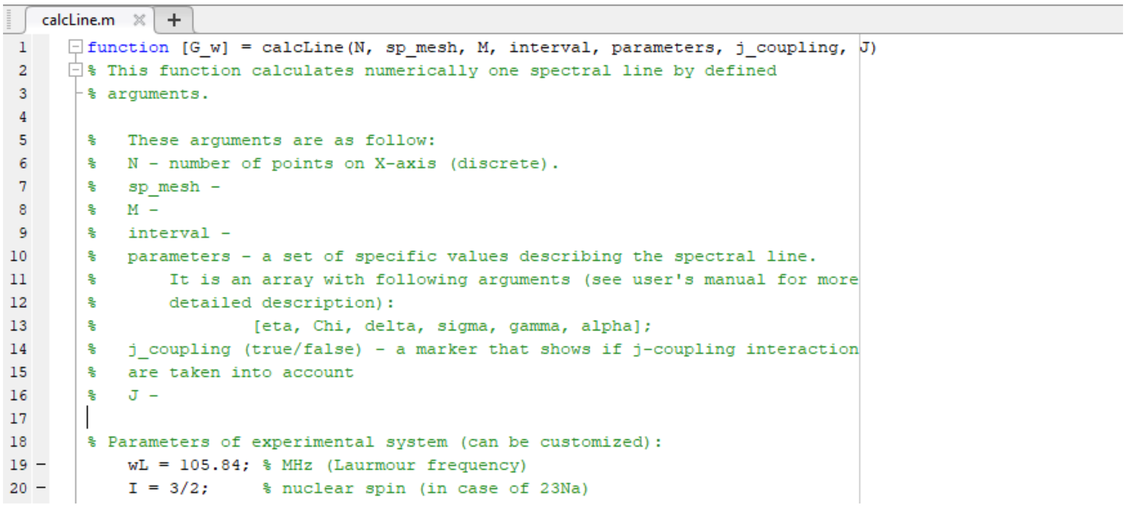
In this case, on 23th iteration error of approximation is ***eps=0.0150*** At the same time here are the following ratios of lines: ***0,454*** (line 1) and ***0,546*** (line 2) , their parameters are:

1) ***eta*** = 0, ***Chi*** = 1.23, ***delta*** = 11.9, ***sigma*** = 80, ***gamma*** = 96;
2) ***eta*** = 0, ***Chi*** = 1.51, ***delta*** = 13.1, ***sigma*** = 80, ***gamma*** = 75.

Settings

Experiment specification

First of all, it is necessary to input into the program information of experimental spectrum. The parameters of which need to be introduced in **calcLine.m**:



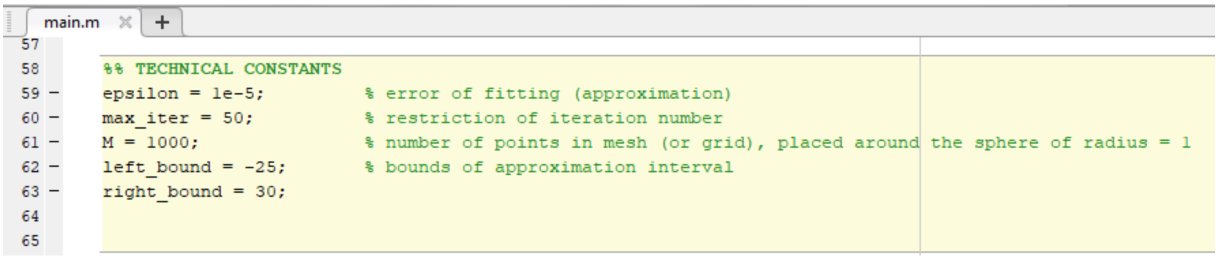
```
function [G_w] = calcLine(N, sp_mesh, M, interval, parameters, j_coupling, J)
% This function calculates numerically one spectral line by defined
% arguments.

%   These arguments are as follow:
%   N - number of points on X-axis (discrete).
%   sp_mesh -
%   M -
%   interval -
%   parameters - a set of specific values describing the spectral line.
%       It is an array with following arguments (see user's manual for more
%       detailed description):
%               [eta, Chi, delta, sigma, gamma, alpha];
%   j_coupling (true/false) - a marker that shows if j-coupling interaction
%   are taken into account
%   J -

% Parameters of experimental system (can be customized):
    wL = 105.84; % MHz (Laurmour frequency)
    I = 3/2;     % nuclear spin (in case of 23Na)
```

wL – Laurmor signal frequency of nucleus, for which experimental spectrum have been obtained, in the magnetic field of the coil. It is set in MHz.

I – spin value of that nucleus.

Technical constants

Here are presented recommendation regarding choice of parameters in section «**Technical constants**» in the file **main.m**.

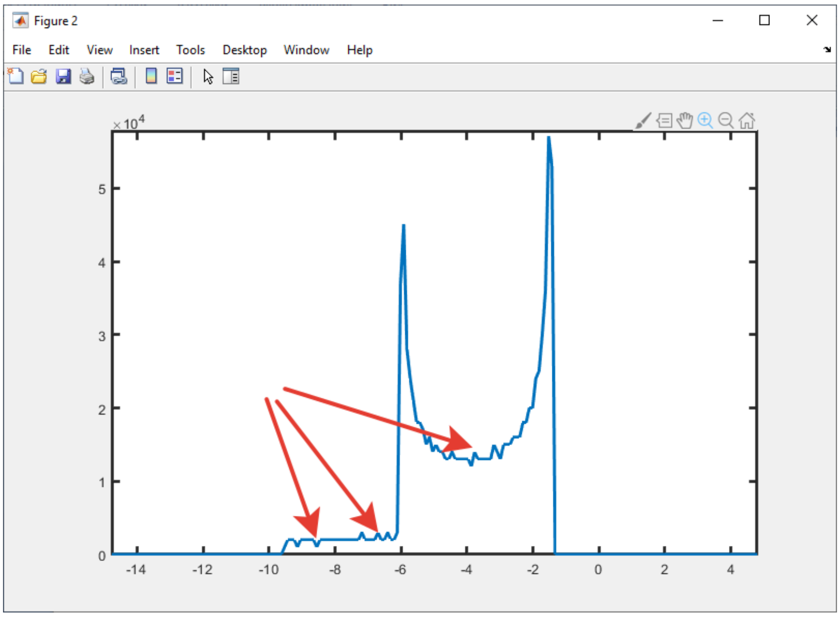


```
%% TECHNICAL CONSTANTS
epsilon = 1e-5;          % error of fitting (approximation)
max_iter = 50;           % restriction of iteration number
M = 1000;                % number of points in mesh (or grid), placed around the sphere of radius = 1
left_bound = -25;        % bounds of approximation interval
right_bound = 30;
```

epsilon – acceptable absolute error of fitting. Distinguishing values: 0,01÷0,00001.

max_iter – a number of iteration steps by each varying parameter of spectrum in gradient descending method. In view of the large time costs, it is better to set values no more than ~100.

M – a number of grid nodes on a unit sphere for accounting equally probable orientations of monocrystals. By default, set value (***M=1000***) is not optimal and does not require correction. Increasing that parameter is acceptable in case of very small width of line from particular monocrystal (width **<10 Hz**). So in the case of the width of line is infinitely small (signal represents a delta-function), the spectrum has a specific inaccuracies:



The spectrum fluctuations, that marked by red arrows, are caused by an error of calculation. It turns out, that amplitude of that disturbances is described as A ~ 1/M.

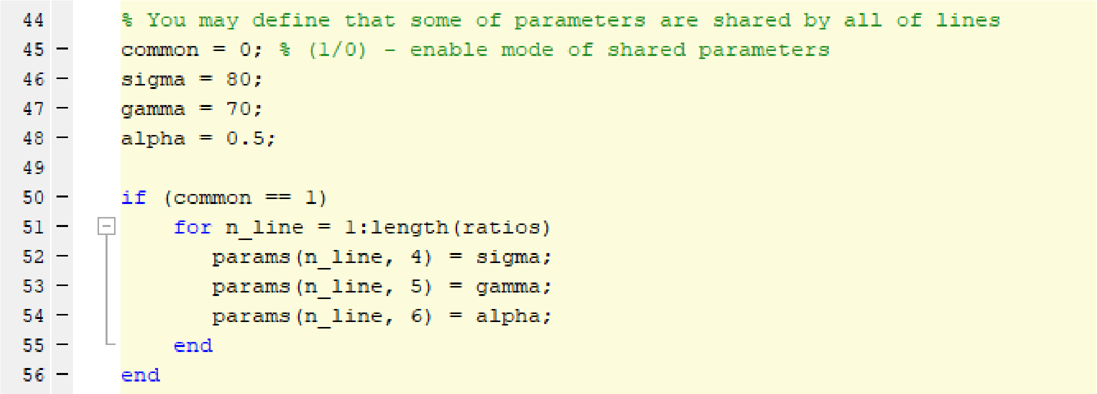
The final spectrum is obtained by convolution of this spectrum on the figure with a function, describing a form of line for particular monocrystal. The convolution of these functions smoothes disturbances if the width of a signal for line is enough great, so that difference between these specta, whether they are built with M=1000 or M=500, almost is not noticeable.

left_bound and ***right_bound*** have a double destination. These set the bounds not only an interval within which approximation is being done, but also another interval where a spectrum graph, windowed in another place, is plotted it should set values of these variables so that the interval covers all the central lines of the calculated spectrum, excluding side bands, appeared under magic angle spinning of the sample.

Shared parameters mode

In case of calculated spectrum is described by several quadrupolar contributions, spectrum approximation might be quite slow. To speed up fitting process, you may use the shared parameters mode. It is convenient if widths of lines relative to different contributions can be similar values.

Below is presented the section of shared parameters (file **main.m**):



```
% You may define that some of parameters are shared by all of lines
common = 0; % (1/0) - enable mode of shared parameters
sigma = 80;
gamma = 70;
alpha = 0.5;

if (common == 1)
    for n_line = 1:length(ratios)
       params(n_line, 4) = sigma;
       params(n_line, 5) = gamma;
       params(n_line, 6) = alpha;
    end
end
```

Description of important parameters:

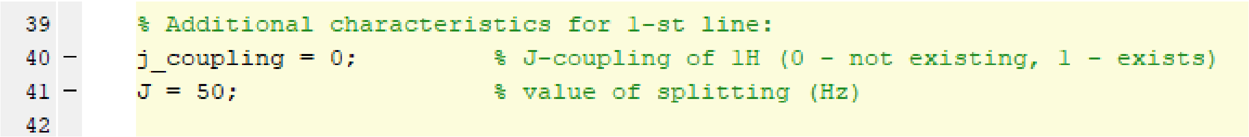
common – a flag indicating of shared parameters mode. It may be of two values: ***0*** (mode on) or ***1*** (off).

sigma, ***gamma***, ***alpha*** – parameters of Gaussian and Lorenz lines, and ratio between them. These specify the waveform of a single crystal in all quadrupole contributions.

J-coupling

In case of magic angle spinning frequency is not enough high, когда частота вращения под магическим углом недостаточно большая, indirect spin-spin interactions can appear in the spectrum. The program allows to set up a value of splitting just for the first spectrum line (the first contribution described in **params** array) caused by that interactions for nuclei that have spin = 1/2.

J-coupling section in **main.m**, «**Initial parameters/constants**»:



```
% Additional characteristics for 1-st line:
j_coupling = 0;         % J-coupling of 1H (0 - not existing, 1 - exists)
J = 50;                 % value of splitting (Hz)
```

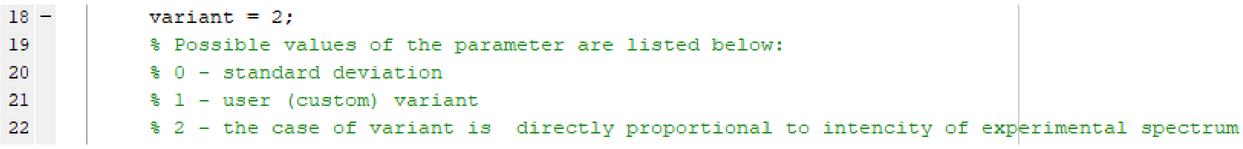
Here:

j_coupling – a flag of j-coupling (a split accounting) mode. The flag may be: ***0*** – (mode off), ***1*** (on).

J – a value of the split, measured in Hertz.

Approximation error

The purpose of fitting is to minimize the deviation (error) between of the spectra. In the program are available several ways of its calculation. The choice of the way is specified in file **calcDiscrepancyWeighted.m**:



```
variant = 2;
% Possible values of the parameter are listed below:
% 0 - standard deviation
% 1 - user (custom) variant
% 2 - the case of variant is  directly proportional to intencity of experimental spectrum
```

variant – a flag, that indicates a calculation method. It can be of three values:

0 – standard deviation:

$$\varepsilon = \sqrt{\sum_{i=1}^{N}(a_i - b_i)^2},$$

where $\{a_i\}_{i=1}^{N}$ – samples of intensities for a calculated spectrum, and $\{b\}_{i=1}^{N}$ – – samples of intensities for an experimental spectrum.

1 – custom option (only for experienced). The option is provided for users convenience of setting formula to be used in calculations.

2 – (*selected by default*) standard deviation with a weight proportional to the intensity of the experimental spectrum:

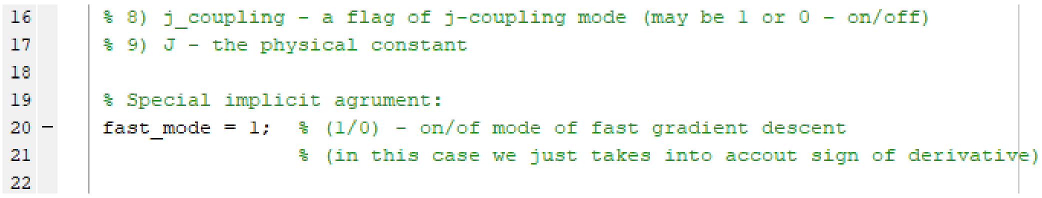
$$\varepsilon = \sqrt{\sum_{i=1}^{N} b_i(a_i - b_i)^2},$$

where the designations are the same as the previous ones.

Gradient descent speed

Fitting of calculated spectrum that consists several quadrupolar contributions (lines) takes a lot of time. To reduce approximation duration, you may use modified gradient descent method.

That method is defined by a flag in files **stepLineParameter.m**, **stepCommonParameter.m** and **StepRatios.m**. In these files the flag is to set right after the function description comments:



```
% 8) j_coupling - a flag of j-coupling mode (may be 1 or 0 - on/off)
% 9) J - the physical constant

% Special implicit agrument:
fast_mode = 1;  % (1/0) - on/of mode of fast gradient descent
                % (in this case we just takes into accout sign of derivative)

```

fast_mode – the flag of gradient descent mode. It may be of 2 values:

0 – «classical» gradient descent:

$$p_k^{(i+1)} = p_k^{(i)} - h_{p_k} \cdot \frac{\varepsilon(p_1, \dots, p_k + \Delta p_k, \dots, p_n) - \varepsilon(p_1, \dots, p_k, \dots, p_n)}{\Delta p_k},$$

where: $p_k^{(i)}$ – value of parameter p_k of k line on i iteration, h_{p_k} – value of step by that parameter, Δp_k – its increment.

1 – «fast» gradient descent (takes into account sign of derivative by parameter):

$$p_k^{(i+1)} = p_k^{(i)} - h_{p_k} \cdot sgn\left(\frac{\varepsilon(p_1, \dots, p_k + \Delta p_k, \dots, p_n) - \varepsilon(p_1, \dots, p_k, \dots, p_n)}{\Delta p_k}\right),$$

Designations are the same as above.

When the calculation takes into account only the sign of the derivative with respect to the parameter, the fit error increases, which is proportional to h_{p_k}. On the other hand, speed of approach optimal value of parameter, at which minimal error is achieved, is also proportional to h_{p_k}. That speed is much higher than one in usual (“classic”) method where it would proportional to $h_{p_k} \cdot \frac{\partial \varepsilon}{\partial p_k}$.

For advanced users

Convenient approach of spectrum approximation

This section describes a method that will help you save time when fitting the spectrum.

Firstly, you should not try too hard to set excellent initial parameters for calculation of the test spectrum before starting the fit. It is enough to specify the values of the parameters at which the calculated and experimental spectra coincide qualitatively.

Secondly, do fitting by choosing the option of the fast gradient descent method. The process will go much faster.

Third, limit yourself to dozens of iterations at given parameter steps and their increment in partial derivatives. The fast gradient descent method is characterized by an error proportional to h_{p_k} (***hValue***). So after 10-20 iterations the method leads to local minimum characterized by error of h_{p_k}, that may be high. Next, to make approximation of calculated spectrum to experimental one, it is need to reduce ***hValue*** (as well as ***dValue***) and to launch fitting process for several dozens of iterations.

I.e. mechanism of you action has to be as follows. Make iterations - > reduce hValue, dValue -> iterate over and so on. As a rule, you have to change the values of steps no more than 1-2 times.

Customization of functionality

The entire source code of the program, if possible, is accompanied by comments from the developer. For functions, a detailed description is provided in the comments. Explanations are made in difficult areas of the program.

Thus, a bet is made that it will be easier for users to modify the source code to fit their needs. A complete list of all files in the software package with a brief description of their main functions is presented below:

main.m	– main file (here are almost all of initial parameters)
calcSpectrum.m	– calculation of a whole spectrum
calcLine.m	– calculation of one spectrum line (contribution)
calcY.m	– calculation of numerical spectrum
calcIndex.m	– calculation of the index for array according to scale
convertToIndices.m	– translation ppm into the horizontal grid index.
calcDiscrepancyWeighted.m	– approximation error calculation
calcDiscrepancy.m	– simple error calculation (not active)
convolution.m	– convolution of functions
integrate.m	– calculates the area under the spectrum
lorenzian.m	– describes the Lorentz function
medium.m	– search for median value
normalizeSpectrum.m	– integral intensity normalization
shiftSpectrum.m	– shifts the spectrum by a given number in ppm
stepLineParameter.m	– step along line-specific parameter
stepCommonParameter.m	– step along all shared parameters
StepRatios.m	– steps along line intensities

Additional notes

Requests for modernization and improvement of the program and this manual can be sent by e-mail *aleksey.kaz@mail.ru* to the author – Kazarinov Aleksey.

Testing this guide and program helped Khasanov Magomed.