User's guide for EvAX 6.16

The most recent version of this manual and EvAX software is available at www.dragon.lv/evax

Janis Timoshenko

Evolutionary Algorithm for EXAFS analysis

JANIS.TIMOSHENKO@GMAIL.COM

WWW.DRAGON.LV/EVAX

If you want to use EvAX code and/or some parts of this document in your work, you are most welcome, but, please cite is as

J. Timoshenko, A. Kuzmin, J. Purans, EXAFS study of hydrogen intercalation into ReO_3 using the evolutionary algorithm, J. Phys.: Condens. Matter 26 (2014) 055401 (15pp).

Last updated October 2023

Contents

1. Introduction

1.1 What is EvAX?

EvAX (Evolutionary Algorithms for XAS analysis) is a program for the analysis of extended X-ray absorption fine structure (EXAFS) data[1] using reverse Monte Carlo (RMC)[2, 3] and evolutionary algorithm (EA) [4] methods. It is designed to be especially useful for studies of materials with well defined structure, such as crystalline and nanocrystalline materials. Some of the main strengths of EvAX method are:

- **Analysis of multiple scattering effects:** to benefit from the structural information, hidden in the contributions of multiple-scattering (MS) effects and that of distant coordination shells, EvAX code provides flexibility for treating all important scattering paths with user-specified precision.
- **Evolutionary algorithm-based structure optimization:** the optimization process that employs the power of evolutionary algorithm (EA) allows more efficient exploration of the possible configuration space and makes feasible advanced analysis of complex compounds with low symmetry.
- **Usage of wavelet transform for EXAFS signal comparison:** the representation of EXAFS spectra in *k*- and *R*-spaces simultaneously using continuous Morlet wavelet transform or Gabor transform allows one to obtain more information from the same experimental data and to have much better control over the difference between the experimental and calculated EXAFS data.
- **Analysis of EXAFS data from several absorption edges simultaneously:** EvAX code allows one to construct an unambiguous single structure model, consistent with EXAFS data, acquired at several absorption edges.

EvAX can be used both with Windows (Windows 7 and newer) and Linux operating systems. The most time-consuming parts of calculations can be automatically divided into smaller problems and performed in parallel (in different threads of execution), taking advantage of multi-processor systems.

1.2 When to use EvAX?

EvAX code is designed for the RMC-like analysis of well-defined structures. Unlike it is in some other implementations of RMC-method, EvAX allows only small atomic displacements of atoms from their initial positions. Hence, it enables the modelling of complex bond-length distributions and allows one to analyze the changes in interatomic distances and structural disorder, but does not allow one to refine the coordination numbers: the latter are imposed by the initial structure model provided by user, and are not changed during the EvAX simulations. Therefore EvAX is useful for cases, where coordination numbers are known in advance, e.g., for the analysis of disorder in crystalline materials or well-defined nanocrystalline materials.

Here are two possible scenarios, when EvAX can be useful:

- You know the approximate structure of the material (e.g., from X-ray diffraction), and you want to use EXAFS data to study the disorder effects and atomic thermal motion.
- You have several plausible structure models for your material (e.g., from DFT simulations), and you want to know, which of them fits better the available experimental EXAFS data.

Besides its main purpose - RMC-like analysis - EvAX can also be used as an efficient tool for calculations of theoretical EXAFS spectra for given low-dimensional atomic configuration or a set of such configurations, generated, e.g., by some molecular dynamics routine (see Sec. 4.7). It can also be used for efficient calculations of wavelet transforms, Fourier transform and inverse Fourier transforms for given EXAFS spectra (see Sec. 5.6), as well as to generate input files for FEFF calculations (see Sec. 4.1).

1.3 What do you need to run EvAX?

EvAX is a command-line program. For calculations following files and programs should be prepared:

1. **EvAX program**, compiled for a given operating system.[1]
2. **FEFF program**[5] (version 8 or higher), compiled for a given operating system;[2]
3. For calculations within Linux OS - the auxiliary **run-FEFF.exe script file** (see Section 4.1);
4. **Parameter file** - text file that contains the values of the parameters for calculations
5. **Structure file** - text file that defines the lattice and types of atoms, and the initial positions of atoms in the structure. Software like VESTA program [6] can be very helpful in preparation of this file.
6. File(s) with experimental extracted **EXAFS data** with subtracted background. EvAX can work directly with EXAFS data, preprocessed by such codes as ATHENA [7] or EDA [8].

The names and formats of structure file, files with experimental EXAFS data, and path to FEFF program are specified within the parameter file. The name of parameter file is supplied to EvAX program as a command-line argument. Thus, to start calculations, just type

```
EvAX.exe    parameters.dat
```

[1] To compile EvAX source code for the given operation system, one should use proper preprocessor directives (*#define*) in the *mysystem.h* file: to choose the operating system either keyword WINDOWS or LINUX should be defined.

[2] FEFF code can be ordered and downloaded from http://monalisa.phys.washington.edu/index-feffproject.html. FEFF version 8.5L is available free of charge and is sufficient for EvAX calculations. Note that default number of atoms allowed in FEFF-8.5L (1000) may be insufficient. See Section 4.1 for details how to fix it.

For Linux OS, of course, slightly different commands needs to used:

```
./EvAX.exe    parameters.dat
```

It is strongly recommended to direct the output of EvAX calculations to a text file (log file), so that the progress of calculations can be recorded. For this purpose use ">" sign at the end of the command, after which the name of the log file follows:

```
EvAX.exe    parameters.dat > logfile.log
```

An example of parameter file can be generated using following command:

```
EvAX.exe    -example
```

This will generate a file "*example.dat*" with the default values for all calculation parameters. **-example** is not the only one additional command that can be supplied to EvAX via command line. Other commands are discussed below. A brief description of these commands (flags) can also be obtained by running

```
EvAX.exe    -help
```

1.4 How does EvAX work?

Nowadays there are a number of *ab-initio* codes available that, knowing the structure and dynamics of the system, allow one to calculate the corresponding theoretical EXAFS spectrum (FEFF [5] and GNXAS [9] codes are the most popular). The agreement between the results of such calculations and experimentally measured EXAFS data usually is good [1]. The problem that we are interested in, however, usually is an inverse problem: we want to find such model of material structure and lattice dynamics that the corresponding calculated EXAFS spectrum is as close to the experimental data as possible.

In the conventional EXAFS analysis the model is set by specifying the distributions of distances between the absorbing atom and its closest neighbours. Usually it is assumed that these distributions are described by Gaussian functions, hence the only variables of such model are the average interatomic distances and the variances of interatomic distances (MSRD, i.e., mean-square relative displacement factors).

In the RMC-like analysis, in turn, to benefit from the constraints set by geometrical considerations, the structure model is formulated directly in the terms of atomic coordinates: the variables are just x, y and z coordinates of each atom in the simulated cell. In the case of crystalline material such cell is often called a "supercell", since it consists of several unit cells. To avoid the surface-related effects the periodic boundary conditions (PBC) can be applied. The supercell should be chosen large enough to represent the structure of a material, but as small as possible to reduce the number of variables in our model. The interatomic distances and MSRD factors can be estimated afterwards from the sets of obtained coordinates for atoms within the supercell.

The original Reverse Monte Carlo (RMC) method is a simulation technique, proposed by McGreevy and Pusztai in the late 1980's, to determine a 3D model of the atomic structure of material by minimizing the difference between its structure-related experimental and calculated properties [10]. In the RMC-type approaches the displacements of atoms are determined by a random process with the aim to minimize the disagreement between calculated and experimentally measured EXAFS data [2].

RMC simulation starts with an arbitrary initial configuration of atoms in a cell of chosen size and shape with or without periodic boundary conditions, for which the total EXAFS spectrum $\chi_{\text{tot}}(k)$ is calculated. Next, the atomic configuration is modified by randomly changing the coordinates of one or several atoms, thus producing a new atomic configuration, for which the total EXAFS spectrum $\chi_{\text{tot}}^{\text{new}}(k)$ is calculated. The two calculated EXAFS spectra $\chi_{\text{tot}}^{\text{old}}(k)$ and $\chi_{\text{tot}}^{\text{new}}(k)$ are compared with the experimental one $\chi_{\text{exp}}(k)$, and the new atomic configuration is either accepted or discarded depending on the results of this comparison. The procedure is repeated as many times as needed until the atoms in the cell occupy such positions that the difference ξ between theoretical $\chi_{\text{tot}}(k)$ and experimental $\chi_{\text{exp}}(k)$ EXAFS spectra is minimized.

Thus five main blocks of such analysis can be identified:

- Preprocessing of experimental EXAFS data;
- *Ab-initio* calculations of theoretical EXAFS spectra for given structure model;
- Algorithms for comparison of experimental and theoretical EXAFS data.
- Structure model optimization, guided by the difference in experimental and theoretical EXAFS data;
- Analysis of the final structure model to extract relevant structural information.

Several challenges associated with such analysis need to be emphasized here. First of all, note that the problem we are trying to solve (i.e., the local structure of sample) is an essentially high-dimensional problem, and to completely characterize the investigated system an enormous amount of variables is required; at the same time the amount of available information in the experimental data is limited. Next, EXAFS spectra that we are trying to interpret depend on the variables of our model (atomic coordinates) in a strongly non-linear way. As a result, the difference (Euclidean distance) between the calculated and experimental EXAFS data is a function of model variables that has many local minima, while we are interested in the global minimum only.

The reverse Monte Carlo/evolutionary algorithm (RMC/EA), implemented in EvAX code, is designed to minimize the influence of these issues. The keys to the problem are

- to reduce the number of involved variables and to reduce the size of configurational space to be explored (i.e., to use relatively small supercell and to constrain atomic displacements);
- to use efficient optimization algorithms;
- to employ fast algorithms to calculate theoretical EXAFS spectra as functions of atomic coordinates;
- to implement novel approaches for signal processing to extract from the available EXAFS spectra as much information as possible.

In further sections of this User's guide we give a detailed description of these main blocks.

2. EvAX parameters

2.1 Parameter file

An example of a parameter file is given below, with all parameters set to their default values. File starts with **BEGIN** keyword and ends, unsurprisingly, with **END** keyword. Lines starting with **#** are neglected. The order of parameters in the file is arbitrary. If a parameter is not specified, the default value is used. Therefore there is no need to specify every time all the parameter values.

The complete list of default values will be printed out to file "*example.dat*", which can be obtained by running

```
EvAX.exe     -example
```

```
BEGIN
Title                                  EXAMPLE
################################################################################
########################## STRUCTURE_MODEL_DEFINITION ##########################
################################################################################
              Number_of_structures                  1
              Scale_coordinates_by                  1
              Initial_displacement                  0.1
              Weight                                1
              Doping_level                          -1
              File_with_structure                   structure.p1
              Supercell                             4x4x4
              Doping                                none
              pbc                                   1
              labels_to_element_name                0
################################################################################
######################### STRUCTURE_MODEL_OPTIMIZATION #########################
################################################################################
              Froze_in                              1500
              Stop_after                            -1
              Number_of_states                      32
              Maximal_step_length                    0.005
              Weight_increment                       0.0001
              Acceptance_rate                        0.8
              Energy_weight                          -1
              Move_type                              move_all
              MT_seed                                291 564 837 1110
              adjust_acceptance                      1
              adjust_weights                         0
              Maximal_displacement                   0.4
              Potential_type                         box
###############################################################################
############################### EXPERIMENTAL_DATA #############################
###############################################################################
              Number_of_spectra                      1
              Columns_in_input                       2
              Column_for_k                           2
              Column_for_chi                         1
              S02                                    1
              dE0                                    0
              k_power_in_I0                          2
              Spectrum_weight                        -1
              File_with_experimental_EXAFS_signal    expEXAFS.dat
              adjust_spectra_weights                 1
###############################################################################
############################### THEORETICAL_DATA ##############################
###############################################################################
              Update_basis_every_...                 2
              Maximal_number_of_clusters             10000
              Expansion_accuracy                     -1
              FEFF_path                              feff85L.exe
              N_legs                                 8
              R_max_for_FEFF                         6
              FEFF_criteria                          0
              Pathfinder_robustness                  0.5
              Clustering_precision                   0.001
              Edge                                   Ge
              Edge_type                              K
              Polarization                           none
              r_scf                                  -1
              ca                                     0.1
              n_scf                                  30
              n_mix                                  1
###############################################################################
####################### THEORY_AND_EXPERIMENT_COMPARISON ######################
###############################################################################
              Space                                  w
              WT_type                                1
              k_min                                  0.05
              k_max                                  19
              dk                                     0.05
              k_power                                2
              FT_window_type                         -1
              R_min                                  0
              R_max                                  5
              R_max_for_1st_shell                    -1
              R0_for_WT                              -1
```

```
###############################################################################
#################################### OUTPUT ###################################
###############################################################################
              time1                                  10
              time2                                  100
              File_for_residuals                     output.dat
              File_for_restart                       restart
              RDF_bin_width                          0.01
              File_for_i-pol._exp._signal            ipolEXAFS.dat
              File_for_BFT_of_exp._signal            expBFT.dat
              File_for_FT_of_exp._signal             expFT.dat
              File_for_WT_of_exp._signal             expWT.dat
              File_for_calculated_EXAFS_signal       EXAFS.dat
              File_for_FT                            ft.dat
              File_for_BFT                           bft.dat
              File_for_WT                            wt.dat
              File_for_partial_EXAFS                 partialEXAFS.dat
              File_for_partial_FT                    partialFT.dat
              File_for_partial_WT                    partialWT.dat
              File_for_partial_RDFs                  rdf.dat
              File_for_final_xyz                     final.xyz
              File_for_equilibrium_xyz               start.xyz
###############################################################################
################################ PARALELIZATION ###############################
###############################################################################
              Number_of_threads                      12
###############################################################################
END
```

2.2 Parameter file for simultaneous fit of several spectra

In the EvAX program it is also possible to analyze several EXAFS spectra (acquired, e.g., at different absorption edges) simultaneously to find a structure model that describes well the available EXAFS data from absorbing atoms of different types.

Number of experimental spectra is specified as **Number_of_spectra** parameter in the parameter file. For example,

Number_of_spectra **2**

results in EvAX trying to open two files with experimental data for further analysis.

If different values of some parameter should be used for the analysis of different spectra, one can just list them all in the line corresponding to this parameter. If only one value is given, it will be applied for the analysis of both spectra. An example of parameter file for the analysis of two spectra simultaneously is given below:

```
BEGIN
Title                                  TWO_SPECTRA
###############################################################################
########################## STRUCTURE_MODEL_DEFINITION #########################
###############################################################################
              Scale_coordinates_by                   1
              Initial_displacement                   0.01
              File_with_structure                    PdAu.p1
              Supercell                              3 3 3
              pbc                                    1
###############################################################################
######################### STRUCTURE_MODEL_OPTIMIZATION ########################
###############################################################################
              Froze_in                               3200
              Stop_after                             -1
              Number_of_states                       64
              Maximal_step_length                    0.005
              Maximal_displacement                   0.5
              Target_dicarding_rate                  0.5
              Potential_type                         box
              Move_type                              move_all
              File_with_potential                    constraints.dat
              MT_seed                                291 564 837 1110
              adjust_acceptance                      1
###############################################################################
############################### EXPERIMENTAL_DATA #############################
###############################################################################
              Number_of_spectra                      2
              Columns_in_input                       2 2
              Column_for_k                           2 2
              Column_for_chi                         1 1
              S02                                    0.86 0.92
              dE0                                    1.91 -3.44
              k_power_in_I0                          2 2
              File_with_experimental_EXAFS_signal    Pd_EXAFS.dat Au_EXAFS.dat
###############################################################################
############################### THEORETICAL_DATA ##############################
###############################################################################
              Update_basis_every_...                 3
              Maximal_number_of_clusters             10000
              Clustering_precision                   0.03
              Expansion_accuracy                     -1
              FEFF_path                              feff85L_jt.exe
              N_legs                                 2 2
              R_max_for_FEFF                         7 7
              FEFF_criteria                          1 1
              Pathfinder_robustness                  0.5 0.5
              Edge                                   Pd Au
              Edge_type                              K L3
              Polarization                           none none
              r_scf                                  -1 -1
              ca                                     0.1 0.1
              n_scf                                  30 30
              n_mix                                  1 1
###############################################################################
####################### THEORY_AND_EXPERIMENT_COMPARISON ######################
###############################################################################
              Space                                  w
              k_min                                  0.05 0.05
              k_max                                  19.95 19.95
              dk                                     0.05 0.05
              k_power                                2 2
              FT_window_type                         -1 -1
              R_min                                  0.01 0.01
              R_max                                  6 6
              R_max_for_1st_shell                    -1 -1
              R0_for_WT                              3.005 3.005
################################################################################
#################################### OUTPUT ####################################
################################################################################
              time1                                  10
              time2                                  100
              File_for_residuals                     output.dat
              File_for_restart                       restart
              File_for_final_xyz                     final.xyz
              File_for_equilibrium_xyz               start.xyz
              RDF_bin_width                          0.01 0.01
              File_for_i-pol._exp._signal            Pd/ipolEXAFS.dat Au/ipolEXAFS.dat
              File_for_BFT_of_exp._signal            Pd/expBFT.dat Au/expBFT.dat
              File_for_FT_of_exp._signal             Pd/expFT.dat Au/expFT.dat
              File_for_WT_of_exp._signal             Pd/expWT.dat Au/expWT.dat
              File_for_calculated_EXAFS_signal       Pd/EXAFS.dat Au/EXAFS.dat
              File_for_FT                            Pd/ft.dat Au/ft.dat
              File_for_BFT                           Pd/bft.dat Au/bft.dat
              File_for_WT                            Pd/wt.dat Au/wt.dat
              File_for_partial_EXAFS                 Pd/partialEXAFS.dat Au/partialEXAFS.dat
              File_for_partial_RDFs                  Pd/rdf.dat Au/partialEXAFS.dat
################################################################################
################################ PARALELIZATION ################################
################################################################################
              Number_of_threads                      2
################################################################################
END
```

Note, if more than one value is specified for some parameters in the parameter file, but **Number_of_spectra** is set to 1, only the first value will be used for calculations, and all the other values will be discarded.

2.3 Editing the parameters

The most straightforward way to modify parameters for EvAX calculations is to modify the parameter file using any text editor. However, one can also modify the parameter file on-line using EvAX. This is convenient in the cases where calculations are carried out remotely. To edit the existing parameter file "*parameters.dat*", use the following command:

```
EvAX.exe    parameters.dat -edit
```

This will start a command line-based interface that will allow viewing and modifing the parameters in the parameters file. Note that the existing "*parameters.dat*" file will not be modified: the modified parameters fill be saved as a new file with the name, e.g., "*parameters.dat_new*".

2.4 List of parameters

2.4.1 Parameters to define structure model

Parameter	Sections in manual
Scale_coordinates_by	6.1
File_with_structure	6.1, 6.2
Supercell	6.1
pbc	6.1
labels_to_element_name	6.1
Initial_displacement	6.4
Doping	6.3
Doping_level	6.3

2.4.2 Parameters for structure model optimization

Parameter	Sections in manual
Move_type	6.5
Maximal_step_length	6.5
Potential_type	6.6
File_with_potential	6.6
Maximal_displacement	6.6
MT_seed	6.7
adjust_acceptance	6.8, 6.9
Acceptance_rate	6.8
Froze_in	6.9, 6.11
Number_of_states	6.9
Stop_after	6.11
time2	6.11

2.4.3 Parameters to describe experimental data

Parameter	Sections in manual
Number_of_spectra	2.2
File_with_experimental_EXAFS_signal	3
Columns_in_input	3
Column_for_k	3
Column_for_chi	3
k_power_in_IO	3
S02	3
dE0	3

2.4.4 Parameters to control the calculations of theoretical data

Parameter	Sections in manual
FEFF_path	4.1
Edge	4.2
Edge_type	4.2
Polarization	4.3
r_scf	4.2
n_scf	4.2
n_mix	4.2
ca	4.2
Pathfinder_robustness	4.4
R_max_for_FEFF	4.5
N_legs	4.5
FEFF_criteria	4.5
Clustering_precision	4.6
Maximal_number_of_clusters	4.6
Update_basis_every_...	4.6
Expansion_accuracy	4.6

2.4.5 Parameters to compare experiment and theory

Parameter	Sections in manual
Space	5.1, 5.2, 5.3, 5.4
k_power	5.2, 5.3, 5.4
k_min	3, 5.2, 5.3, 5.4
k_max	3, 5.2, 5.3, 5.4
dk	3, 5.1
FT_window_type	3, 5.1
R_min	5.3, 5.4
R_max	5.3, 5.4
R0_for_WT	5.4
WT_type	5.5
R_max_for_1st_shell	3
adjust_spectra_weights	5.7
Spectrum_weight	5.7

2.4.6 Parameters to define output

Parameter	Sections in manual
File_for_equilibrium_xyz	7.1
File_for_final_xyz	7.1
File_for_i-pol._exp_signal	7.2
File_for_FT_of_exp_signal	7.2
File_for_BFT_of_exp_signal	7.2
File_for_WT_of_exp_signal	7.2
File_for_FT	7.2
File_for_BFT	7.2
File_for_WT	7.2
File_for_calculated_EXAFS_signal	7.2
File_for_partial_EXAFS	7.2
File_for_residuals	7.3
File_for_restart	7.4
time2	7.1, 7.2, 7.4
time1	7.2, 7.3
Column_for_k	3, 7.2
Column_for_chi	3, 7.2
k_power_in_IO	3, 7.2
File_for_partial_RDFs	7.5
RDF_bin_width	7.5

2.4.7 Parallelization parameters

Parameter	Sections in manual
Number_of_threads	2.5

2.5 Parallelization

In EvAX simulations, the most time-consuming parts of calculations (grouping of similar scattering paths, *ab-initio* EXAFS calculations, wavelet transform of EXAFS spectra) can be automatically divided into smaller problems and performed in parallel (in different threads of execution), taking advantage of multi-processor systems. For this purpose for LINUX version of the program the *pthreads* library is used, and for WINDOWS version - *CreateThread()* and related functions from the standard Windows library *windows.h*.

Number of threads to be used in EvAX calculations is specified by **Number_of_threads** parameter in parameters file. For example,

```
Number_of_threads    8
```

will split some steps of EvAX calculations in 8 tasks. If 8 CPU cores are available, each task will be then delegated by operating system to a separate core. (**Number_of_threads** can also be larger than the number of CPU cores available, but, of course, no improvements in calculations speed can be expected in this case).

Note that increasing the number of cores available 2 times will not result in the twofold increase of computational speed. This is because not all steps in EvAX algorithm are parallelized, and also because creation and deletion of threads require additional resources. The optimal number of cores depends on the particular problem, but, in general, there is no signficant improvement in the calculation speed, if more than 16 cores are used.

3. Experimental data

Experimental EXAFS data with subtracted background should be provided for EvAX calculations. The name of the file with experimental data should be supplied as **File_with_experimental_EXAFS_signal** parameter. As an alternative, one can also supply the name of the file with experimental data via the command line argument, using **-spectrum** option:

```
EvAX.exe    parameters.dat -spectrum EXAFS.dat
```

In this case experimental data will be imported from file "*EXAFS.dat*". The parameter **File_with_experimental_EXAFS_signal** in the parameters file will be ignored in this case. Note that this approach to specify the name of file with experimental data cannot be used if two or more spectra are analyzed simultaneously.

By default, EvAX assumes that there are two columns in the data file, the 1st column contains $\chi(k)k^2$ values, while the 2nd column contains k values (file format compatible with EDA package [8]). If the order or number of columns, or the power factor n in the k^n term is different, these correct values should also be specified in the parameters file. For example, with parameters as follows

```
File_with_experimental_EXAFS_signal   EXAFS.dat
Columns_in_input                      4
Column_for_k                          1
Column_for_chi                        3
k_power_in_IO                         0
```

EvAX will try to read experimental EXAFS data from file "*EXAFS.dat*", assuming that it contains 4 columns, k values are given in the first column, while EXAFS values $\chi(k)k^0$ values are given in the third column.

Note that similar format will also be used for output of theoretically calculated EXAFS: in this example the output file will contain 2 columns, where k values are given in the first column, while EXAFS values $\chi(k)k^0$ values are given in the second column. At the same time, note that power factor n used for data input and output is not the power factor that will be used for comparison of theoretical and experimental EXAFS data (the latter is specified by **k_power** parameter.)

Before the main EvAX run, EXAFS data will be re-interpolated on the k-grid with k-values between **k_min** and **k_max** and k-step **dk**, specified in the parameters file. E.g.,

```
k_min    3.0
k_max    15.0
dk       0.05
```

specifies that comparison of theoretical and experimental EXAFS spectra will be carried out on a k-mesh that spans k-range between 3.0 and 15.0 Å^{-1} with the step 0.05 Å.

The experimental data should also be corrected for non-structural parameters ΔE_0, which sets the correction for photoelectron reference energy, and S_0^2 factor, which describes the EXAFS amplitude reduction due to many-electronic effects [1]. It is very important to set these parameters to correct values. Since ΔE_0 value correlate strongly with the values of interatomic distances, and S_0^2 value correlate strongly with the values of disorder factors, inaccurate values of these parameters will result in significant deformations of bond length distributions.

Both ΔE_0 and S_0^2 can be obtained from conventional EXAFS analysis of, e.g., first coordination shell contribution, or from the analysis of suitable reference material. Alternatively, one can perform several EvAX calculations with different ΔE_0 and S_0^2 values, and choose the model that gave the best agreement with experiment.

Note that if ΔE_0 is estimated from the conventional fit with theoretical phases and amplitudes, generated by FEFF code, the same version of FEFF code needs to be used for this analysis and the following EvAX calculations, since different versions of FEFF code may have slightly shifted estimates for E_0 value. Note also that the ΔE_0 value in the EvAX input file should be with *the opposite sign* than yielded by conventional analysis (since we are shifting experimental data rather than model data in this case).

To specify ΔE_0 and S_0^2 values for EvAX calculations, use **S02** and **dE0** parameters. For example,

```
S02    0.8
dE0    -2.5
```

will shift the energy scale by -2.5 eV, and divide experimental signal by 0.8, before it is compared with theoretical data.

Note that EvAX can attempt to estimate E_0 and S_0^2 values automatically. Before the main run, it performs conventional fit of the 1st coordination shell. The results of this fitting (E_0 and S_0^2 values, as well as the average interatomic distance and disorder (MSRD) factor) are printed to the standard output (the obtained E_0 value should be with an opposite sign than that for the value specified as **dE0** parameter). Fitting is performed in the R-range up to $R_{max}^{\text{1st shell}}$. The value of the latter is specified in the parameters file as **R_max_for_1st_shell**. For example,

```
R_max_for_1st_shell    2
```

will set the value of $R_{max}^{\text{1st shell}}$ to 2.0 Å. If **R_max_for_1st_shell** is set to a negative value, a default value of $R_{max}^{\text{1st shell}}$ will be used, equal to the nearest-neighbor distance in the starting structure + 0.1 Å. It is important to remember that for the conventional 1st shell fitting, EvAX provides only very limited possibilities that work only for very simple materials. Therefore one should always critically examine the obtained E_0 and S_0^2 values, and, in case of any doubts, fitting should be done with more specialized software (such as, e.g., FEFFIT or EDA). This is the reason, why by default the E_0 and S_0^2 values, obtained by EvAX, are not used for calculations, and printed out only for user's information. If one, nevertheless, wants to apply these values automatically, additional keyword **-autoES** needs to be specified when launching EvAX:

```
EvAX.exe    parameters.dat    -autoES
```

4. EXAFS calculations

4.1 FEFF

RMC/EA procedure requires to calculate at each iteration the total EXAFS spectrum $\chi_{\text{tot}}(k)$, corresponding to the current atomic configuration. This spectrum then is compared with the experimental one $\chi_{\text{exp}}(k)$. The total EXAFS spectrum $\chi_{\text{tot}}(k)$ is an average of the EXAFS spectra $\chi(k)$ from all absorbing atoms in the atomic configuration. These spectra can be calculated by one of the available ab initio EXAFS codes. In EvAX program we use the self-consistent real space multiple-scattering approach as is implemented in the FEFF code [5, 1]. FEFF code is developed by J. Rehr et. al., for the simulations of XAS and related properties (X-ray magnetic dichroism, X-ray photoemission spectra, etc.) of given atomic cluster. Undoubtedly, currently it is one of the most popular and reliable softwares for EXAFS calculations, which provides a good description of experimental data. See [5] and [11] for more details.

The FEFF program is used as an internal routine in the EvAX code. The input file "*feff.inp*", required for FEFF calculations, is created and the FEFF program is started automatically by EvAX software. The recommended version if FEFF8-*lite*, available free of charge from the FEFF project's webpage http://monalisa.phys.washington.edu/. Note that more recent FEFF version (FEFF9) should also be compatible with EvAX, but it has not been extensively tested.

To use FEFF8-*lite* with EvAX, follow these instructions:

1. Download the Fortran source code "feff85L.f" from http://monalisa.phys.washington.edu//feff/Bin-Source-feff8Lite/Source/feff85L.f
2. By default the number of atoms that can be used for ab-initio EXAFS calculations in FEFF8-*lite* is limited by 1000, and it is not sufficient for a realistic crystalline structure model. To increase the allowed number of atoms for FEFF calculations, edit the "feff85L.f" file: find ALL 130 definitions of parameters `natx` and `nattx`, and set the values of these parameters to, for example, 10000.
3. Use, for instance, Intel *ifort* compiler to create a new FEFF executable.
4. Specify the path and name of your FEFF program in the input file for EvAX calculations.

The exact name and location of FEFF program should be given to EvAX as parameter **FEFF_path**, for example

```
FEFF_path:   feff85L.exe
```

Note for LINUX users: an additional script "*run-FEFF.exe*" file should be added to the directory that contains EvAX executable. This text file contains just three lines

```
#!/bin/bash
cd $1
./feff > feff.log
```

4.2 Self-consistent cluster potential

Let us remind here, the EXAFS spectrum $\chi(k)$ is given by equation

$$\chi(k) = S_0^2 \sum_j \frac{f_j(k,\vec{R}_j)}{kR_j^2} \sin(2kR_j + \phi_j(k,\vec{R}_j)). \tag{4.1}$$

Here the summation is carried out over all possible scattering paths, R_j is the half length of the j-path, and the functions $f_j(k,\vec{R}_j)$ and $\phi_j(k,\vec{R}_j)$ are scattering amplitude and phase shift functions. Both of them depend on the electronic properties (electronic density of states) of absorbing and scattering atoms, but also on photoelectron energy and both radial and angular characteristics of the scattering path (vector $\vec{R}_j$ denotes the position of the atoms, involved in the j-th scattering path, with respect to the absorbing atom).

The calculations of the f_j and ϕ_j functions in the FEFF code require first to calculate the electrostatic potential in atomic cluster that involves the absorbing atom and a few coordination shells around it. Such potential is obtained in self-consistent and quite time-consuming calculations. Luckily, in EXAFS region we usually can neglect the variations of cluster potential due to disorder effects and atomic displacements. Thus in EvAX code the self-consistent cluster potential is evaluated only once (before the main RMC/EA run) for the starting atomic configuration.

In EvAX by default for the calculations of the self-consistent cluster potential the complex exchange-correlation Hedin-Lundqvist potential and default values of muffin-tin radii, as provided within the FEFF code [5], are used. Other parameters for self-consistent calculations can be specified through EvAX parameter file.

Obviously, for EXAFS calculations one needs to specify at least the absorbing atom and the absorption edge. This can be done with EvAX parameters **Edge** and **Edge_type**.

For example,

```
Edge        W
Edge_type   L3
```

will request calculations for tungsten L_3 edge. The name of absorbing atom can also be something like **W0** or **W1** (up to **W9**). Importantly, the label of the absorbing atom should be the same as specified in the structure file. If in the structure file atoms of the same type but with different labels are specified (e.g., both W1 and W2 are used), EvAX will calculate EXAFS spectra only for the atoms, whose labels coincide with the **Edge** parameter value. Clearly, it is an undesirable behavior for most applications (since experimental EXAFS spectrum is, of course, always averaged over all atoms of the same type). Therefore normally one needs to use the same label for all absorbing atoms in the structure file (different labels can be used for non-absorbing atoms, though).

Edge_type parameter, in turn, typically is equal to **K** or **L3**, or, more rarely, **L1** or **L2**. Calculations for M-shells or higher are possible, in principle, but have never been tested.

For calculations of self-consistent potential some other parameters may be important. In particular, the size of the atomic cluster R_{scf} used for self-consistent potential calculations (first parameter in FEFF "SCF" card), is set by **r_scf** parameter. If this value is set to a negative value, a default value equal to the radius of the 3rd coordination shell will be used. The latter, in turn, is estimated by EvAX based on the provided structure model. The default value will work well, if the provided initial structure model corresponds to a simple, relatively symmetric structure.

Note that **r_scf** needs to be at least as large as the distance between the absorbing atom and the nearest atom of each type (i.e., atoms of all types need to be present withing the cluster used for potential calculations).

Maximum number of iterations for the potential calculations is specified by **n_scf** parameter. The default value is 30, which is normally sufficient. The convergence accelerator factor affects the convergence of self-consistent calculations. It can be set by **ca** parameter. The default value 0.1 is suitable for most cases. Finally, **n_mix** parameter specifies how many iterations FEFF will perform with the mixing algorithm before starting the so called Broyden algorithm. Normally it should be equal to 1. All these parameters are used to fill in the "SCF" card of FEFF input file.

For example,

```
r_scf    5.0
n_scf    20
ca       0.2
n_mix    1
```

corresponds to FEFF input

```
   SCF 5.0 20 0.2 1
```

Potential calculations are carried out immediately after EvAX code is started. FEFF input and output files can be found then in directory "feff_0". If calculations for two or more absorption edges are requested, corresponding files will be saved to directories "feff_1", "feff_2", etc. These files will not be overwritten during subsequent EvAX iterations (subsequent FEFF calculations will be carried out in other directories, with the names like "feff_0_0", "feff_0_1", "feff_1_0", "feff_1_1", etc).

If there are any problems with the potential calculations, they normally will be identified in "*log1.dat*" file, produced by FEFF. User should always check that no error or warning messages are printed out to this file. Most problems can be resolved by increasing **n_scf** and reducing **ca** parameters.

In some cases, however, better control over potential calculations is necessary. In these cases it is possible to use for EvAX simulations manually pre-calculated FEFF potentials. In these cases all relevant FEFF output files need to be saved before EvAX calculations in "feff_0" (and "feff_1", "feff_2", ..., if necessary) directory. Next, when starting EvAX calculations, additional option **-skippot** needs to be specified to avoid the recalculation of cluster potential:

```
EvAX.exe    parameters.dat    -skippot
```

An example, where such approach can be useful, is EvAX analysis of nanostructured materials. In this case one may want to use for EvAX calculations the cluster potential, calculated, e.g., for well-defined bulk material.

It is also possible to use EvAX as a convenient tool to generate FEFF input files. To generate FEFF input without launching the RMC calculations, use **-atoms** option:

```
EvAX.exe    parameters.dat    -atoms
```

Here parameter file "*parameters.dat*" needs to specify only the structure file and absorption edge, and, optionally, parameters for self-consistent calculations. As during the normal EvAX run, the created FEFF input file "*feff.inp*" will be saved to "feff_0" directory.

For more details on self-consistent potential calculations we recommend to refer to FEFF manual [12].

4.3 Polarization-dependent XAS

Sensitivity of EXAFS to a particular bond depends on the angle between the bond and the polarization of incoming X-rays. When performing XAS experiments for polycrystalline samples or liquids, this effect of X-ray polarization averages out. However, for oriented samples it is possible to use it to probe directly anisotropy of material's structure. To carry out EXAFS calculations for a particular X-ray polarization, use **Polarization** parameter, followed by three numbers (separated by underscore) that specify the coordinates of the nonzero polarization vector. This notation is thus similar as used in the FEFF input file for polarization-dependent calculations (except for the underscore). For example,

```
Polarization    1_1_0
```

will carry out EXAFS calculations with X-ray polarization vector in (110) direction. The default value for **Polarization** parameter is **none**, which corresponds to the calculations of polarization-averaged spectra.

Note that if EXAFS spectra collected with different X-ray polarization directions are available, they all can be used simultaneously in RMC simulations, to find a single structure model consistent with all experimental information. The input and output files in this case are organized in the same way as for simultaneous RMC-modeling of spectra collected at different absorption edges, and the corresponding polarization vectors should be given one after another in the **Polarization** field. For example of such calculations see Ref.[13].

4.4 Photoelectron paths and their agglomeration

Let us get back to Eq. (4.1). Even in the distorted structure, where all atoms are displaced from their equilibrium positions, it is convenient to group paths with similar geometry and length.

We can rewrite therefore this equation as

$$\chi(k) = S_0^2 \sum_p \sum_j \frac{f_p(k, \vec{R}_{pj})}{kR_{pj}^2} \sin(2kR_{pj} + \phi_p(k, \vec{R}_{pj})). \quad (4.2)$$

Here p is an index of a group of similar paths. Let's discuss, how such grouping can be carried out.

Note that since only relative positions of scattering atom with respect to the absorbing atom are important, each single scattering path can be uniquely defined with just a single number, corresponding, e.g., to the distance between the absorbing and scattering atoms. For double-scattering paths $n = 3$ parameters are necessary, since to characterize the location of another scattering atom with respect to the absorbing and the first scattering atom two additional degrees of freedom are necessary. For even more complex paths (paths with order $d > 2$) the total number of degrees if freedom n can be calculated as $n = 3 + 3 \cdot (d - 2)$. Thus each path can be represented as a point in n-dimensional space.

Now it is easy to find and group similar paths: two paths will be considered as belonging to the same group, if both paths have the same dimension, and the distance between corresponding points in n-dimensional space is smaller than the value of **Pathfinder_robustness** parameter. E.g.,

```
Pathfinder_robustness    0.5
```

will group paths in one group, if the distance between corresponding points is less than 0.5 Å.

Note that so far Eq. (4.2) is still exact for all, even very large values of **Pathfinder_robustness** parameter. The value of this parameter, however, will affect the accuracy of approximations, described below. Values between 0.1 and 1.0 Å usually work well.

4.5 Paths importance and filtering

The number of possible photoelectron scattering paths grows exponentially with the increase of the maximal paths length $2R_{\text{path}}$. Therefore one needs to consider carefully, which paths will be included in the analysis.

The easiest way how to reduce the number of paths is to reduce R_{path} value. Since for the comparison of experimental and theoretical data we rely on Fourier or wavelet transforms, we can easily filter out the contributions of longer paths from the experimental data, and hence we do not need to include them in theoretical calculations. The maximal length for paths included in the calculations is determined by **R_max_for_FEFF** parameter. E.g.,

```
R_max_for_FEFF    5.0
```

will include in the calculations only paths with total length $2R_{\text{path}}$ smaller than $2 \cdot 5.0$ Å. Importantly, **R_max_for_FEFF** parameter should normally be 0.5 - 1.0 Å **larger** than **R_max** parameter value, which limits the R-range used for the comparison of experimental and theoretical EXAFS data: as a result of spectral broadening due to disorder effects, finite signal length and dependencies of scattering functions on wavenumber, k the paths with the length $2R_{\text{path}}$ have nonzero contribution to EXAFS data in the R-range that is slightly below R_{path} value.

Another way how to reduce the number of paths that is included in the analysis is to limit the maximal allowed path order d. The maximal allowed path order is determined by **N_legs** parameter. E.g.,

```
N_legs    3
```

will limit the analysis to single-scattering and double-scattering paths only. Note, however, that it is usually very hard to say in advance, what is the maximal order for important paths. For example, in cubic materials due to the so called focusing effect the amplitudes of double- and triple-scattering paths can exceed significantly amplitudes of distant single-scattering paths. Therefore we do not recommend to use **N_legs** parameter to reduce the total number of paths.

Much better solution is to make the decision to discard the paths based on their actual importance: the relative amplitude of corresponding partial contribution to the total EXAFS.

EvAX carries out such importance estimation for the initial structure model. After similar paths are grouped together (see previous Section), EXAFS spectrum for initial configuration is calculated. For each of the path groups a partial contribution $\chi_p(k)$ is compared with the the total EXAFS $\chi_{\text{tot}}(k)$. If ε_p, defined as

$$\varepsilon_p = 100 \left(\frac{\int\limits_0^{20} (\chi_p(k) k^n)^2 dk}{\int\limits_0^{20} (\chi_{\text{tot}}(k) k^n)^2 dk} \right)^{1/2}, \tag{4.3}$$

is smaller than the value of **FEFF_criteria** parameter, the corresponding group of paths will not be included in the analysis. Here n is the power factor, used for further EXAFS analysis (**k_power** parameter).

E.g., if

```
FEFF_criteria    1.5
```

is used, groups of paths with total contribution smaller than 1.5% will be neglected. **FEFF_criteria** values between 1.0 and 5.0 usually yield a good compromise between calculation speed and accuracy.

The paths groups that have passed this test and will be used for further analysis are listed by EvAX in the standard output. List of these paths with more details is saved as "*paths_fin.dat*" file in "feff_0" directory (and "feff_1", "feff_2", etc., directories, if more than one EXAFS spectrum is analyzed simultaneously). The structure of this file resembles strongly the structure of "*paths.dat*" file, generated by FEFF program.

If more advanced control over which paths are included in the analysis is needed, user can interrupt EvAX calculations and manually edit "*paths_fin.dat*" file, and remove from it an arbitrary number of paths. It is not possible, however, to add new paths or change their order. After "*paths_fin.dat*" is modified, EvAX calculations can be restarted with additional option **-skippaths**, which protects file "*paths_fin.dat*" from overwriting:

```
EvAX.exe    parameters.dat    -skippaths
```

4.6 Phases and amplitudes

For calculations of scattering phase and amplitude functions we again use FEFF code.

As emphasized in Eqs.(4.1) and (4.2), scattering phases and amplitudes depend not only on wavenumber k, but also on relative arrangement of scattering atoms. For example, well-known is the fact that scattering amplitude for double- and triple-scattering paths is very sensitive to the scattering angle, if this angle is close to 0 degrees (so called focusing effect, which is very pronounced in materials with cubic or nearly-cubic lattice). While this seems to be a significant complication for EXAFS analysis, this fact ensures the sensitivity of EXAFS analysis to the many-body distribution functions, i.e., 3D atomic structure, rather just radial distribution functions. Such information, e.g., is not present in total scattering/PDF data, which is a complementary technique to EXAFS data analysis.

Thus, in principle, one needs to calculate scattering phase and amplitude functions for each of the thousands of possible scattering paths in distorted structure model. This would require a lot of computational resources, and would be inefficient, since both amplitude and phase are relatively smooth functions and in most cases their dependency on paths geometry is relatively weak. Moreover, in a well-ordered structure many scattering paths are very close, and they change only little from one RMC iteration to the next one.

In conventional EXAFS analysis one usually assumes that amplitudes and phases are the same for all similar paths (paths in one paths group). While this approach works reasonably for the analysis of single-scattering contributions, due to the abovementioned issues it is inaccurate for materials, where multiple-scattering effects are important.

Therefore in EvAX following approach is implemented. We further rewrite Eq. (4.2) equation as

$$\begin{aligned}\chi(k) &\approx S_0^2 \sum_p \sum_c^{c_{\max}(p)} \sum_j \frac{f_{pc}(k)}{kR_{pcj}^2} \sin(2kR_{pcj} + \phi_{pc}(k)) = \\ &= S_0^2 \sum_p \sum_c^{c_{\max}(p)} \frac{f_{pc}(k)}{k} \sum_j \left(\frac{\sin(2kR_{pcj})}{R_{pcj}^2} \cos(\phi_{pc}(k)) + \frac{\cos(2kR_{pcj})}{R_{pcj}^2} \sin(\phi_{pc}(k)) \right) = \\ &= S_0^2 \sum_p \sum_c^{c_{\max}(p)} \frac{f_{pc}(k)}{k} \left(\cos(\phi_{pc}(k)) \sum_j \frac{\sin(2kR_{pcj})}{R_{pcj}^2} + \sin(\phi_{pc}(k)) \sum_j \frac{\cos(2kR_{pcj})}{R_{pcj}^2} \right)\end{aligned} \tag{4.4}$$

Here f_{pc} and ϕ_{pc} are scattering functions for p-th paths group, calculated for c-th path from that group. The c-th phase and amplitude will be used to calculate EXAFS contributions for all those paths in the p-th group, for which the c-th path is the closest from all $c_{\max}(p)$ paths with calculated scattering functions.

Clearly, the accuracy of this approximation depends on the number of paths $c_{\max}(p)$ with calculated phases and amplitudes. For example, if we want to calculate spectra as accurate as possible, we set $c_{\max}(p)$ to be equal to the total number of paths in the p-th group. If we want to carry out fast calculations with accuracy as in conventional EXAFS analysis, we need to set $c_{\max}(p)$ to 1. In practice, however, EvAX determines the optimal $c_{\max}(p)$ value for each paths group, dependently on the sensitivity of that path scattering functions to exact path geometry and relative importance of that path in the total EXAFS spectrum.

Each EvAX calculation starts with $c_{\max}(p) = 1$ for all paths groups. Before the main calculations are started, for each paths group EvAX calculates EXAFS spectrum in two different ways: approximately (using Eq. (4.4)), and precisely (using exact phases and amplitudes for each of the scattering paths). The two obtained spectra are then compared, and if the relative difference between them is larger than some small value ε, the $c_{\max}(p)$ is then increased by 1. The process is repeated, until the required accuracy is reached. The value of ε is set by **Clustering_precision** parameter.

```
Clustering_precision    0.05
```

is normally sufficient for reasonably accurate calculations. Note that if multiple-scattering effects are contributing strongly to the signal, it is better to reduce ε value to, e.g., 0.001.

As the material model evolves, it typically gets more distorted, hence the distribution of different paths gets broader. Therefore from time to time EvAX will repeat the accuracy check, and will increase the $c_{\max}(p)$ values, if necessary. The frequency of such accuracy checks is determined by **time2** parameter. E.g.,

```
time2    100
```

indicates that EXAFS calculation accuracy check will be repeated after each 100 iterations. Note that since the accurate EXAFS calculations are quite time consuming, it is better not to set **time2** parameter to some unnecessarily small value. Values between 100 and 500 usually work fine. In addition, if one wants to run calculations without such accuracy check at all, one can use option **-skipcheck**:

```
EvAX.exe    parameters.dat    -skipcheck
```

Note that all $c_{\max}(p)$ values will be equal to their initial values 1 during the whole calculations in this case. Hence it is not recommended to use this option for other than testing purposes.

Maximal total number of paths with calculated phases and amplitudes ($\sum_p c_{\max}(p)$) is determined by **Maximal_number_of_clusters** parameter. For example,

```
Maximal_number_of_clusters    10000
```

indicates that if the total number of paths with calculated phases and amplitudes has reached 10000, further increase of $c_{\max}(p)$ will not happen. Note that the reduction of phases and amplitudes recalculations using **Maximal_number_of_clusters** parameter is not recommended, and should be used in "emergencies" only (e.g., when the number of phases and amplitudes calculated exceeds the memory, available for calculations).

Further increase of computational speed is possible, if one notes that geometries of scattering paths do not change much within one iteration, since the atomic displacements are small. Therefore, if at some moment instead of recalculating phases and amplitudes one will use the phases and amplitudes, calculated for the previous iteration, the corresponding error in theoretical EXAFS data will be small. To specify that phases and amplitudes needs to be recalculated, e.g., every second, or every third iteration, one can use **Update_basis_every_...** parameter. E.g.,

```
Update_basis_every_...    2
```

indicates that phases and amplitudes will be recalculated only at every second iteration. It is not recommended to have the value of this parameter larger than 3.

Finally, after all these approximations are implemented, it could happen that the most time-consuming part of EXAFS calculations will be actually calculation of $\sum_j \frac{\sin(2kR_{pcj})}{R^2_{pcj}}$ and $\sum_j \frac{\sin(2kR_{pcj})}{R^2_{pcj}}$ sums. Calculations can be sped up, if the addends are expanded in Taylor series: the summation then can be carried out analytically. Note that for sinusoidal functions Taylor expansion converges VERY slowly, and many expansion terms may be needed to achieve reasonable accuracy. Fortunately, knowing the length of the spectrum in k-space and maximal expected deviations from the average path length, it is possible to predict the number of expansion terms that will result in necessary expansion accuracy. It can be shown that

$$\begin{aligned}
&\sum_j \frac{\sin(2kR_j)}{R_j^2} \approx A_n(k,\langle R\rangle)\cos(2k\langle R\rangle) + B_n(k,\langle R\rangle) \text{ and} \\
&\sum_j \frac{\cos(2kR_j)}{R_j^2} \approx B_n(k,\langle R\rangle)\cos(2k\langle R\rangle) - A_n(k,\langle R\rangle),
\end{aligned} \tag{4.5}$$

where $2\langle R\rangle$ is the average path length and

$$\begin{aligned}
A_n(k,R) &= \sum_{i=2}^{n} a_i(k,R)\mu_i, \text{ and} \\
B_n(k,R) &= \sum_{i=2}^{n} b_i(k,R)\mu_i,
\end{aligned} \tag{4.6}$$

μ_i is i-th central moment of paths length distribution,

$$\begin{aligned}
a_i(k,R) &= \sum_{j=1}^{i/2} \alpha_{i,2j+1}(k,R), \\
b_i(k,R) &= \sum_{j=1}^{i/2} \beta_{i,2j}(k,R) \\
\alpha_{i,3}(k,R) &= -2ikR(-1/R)^{i+2}, \\
\alpha_{i,j}(k,R) &= -\alpha_{i,j-2}(k,R)(2kR)^2(\alpha - j + 1)/(j(j-1)(i-j+3)), \\
\beta_{i,2}(k,R) &= \frac{(1+i)}{R^2}(-1/R)^i, \\
\beta_{i,j}(k,R) &= -\beta_{i,j-2}(k,R)(2kR)^2(\alpha - j + 1)/(j(j-1)(i-j+3)),
\end{aligned} \tag{4.7}$$

Despite the ugliness of these expressions, summation via Taylor expansion is able to increase the calculation speed significantly (up to 3 - 4 times in some cases!). The accuracy of expansion is determined by the number of n (number of used central moments), and is set by parameter **Expansion_accuracy**. E.g.,

```
Expansion_accuracy    14
```

sets the number n to 14, which is a reasonable value for calculations of EXAFS spectra for systems, where deviations from average path length do not exceed ca 0.5 Å. If larger deviations are expected, one needs to increase the value of **Expansion_accuracy**. If **Expansion_accuracy** is set to some negative value, this approximation is not used and sums on the left side of Eq.(4.5) are calculated exactly, which may be a preferred option for strongly disordered materials. And indication that **Expansion_accuracy** needs to be increased is an unphysical increase in the amplitude of calculated EXAFS at high k values.

4.7 EXAFS for given structure

The power of EvAX to efficiently calculate EXAFS spectra can be used without running RMC simulations: EvAX can calculate corresponding EXAFS spectrum for any atomic configuration, provided by user. Typical example of such application would be calculations of averaged EXAFS spectrum for many thousands of atomic configurations, generated, e.g., with some molecular dynamics routine.

Atomic configurations need to be provided in the XYZ format, as discussed in Sec. 7.1. If many configurations are used, they can be listed one after another in the same XYZ file. To calculate corresponding EXAFS spectrum, one needs to run EvAX with additional keyword **-md**. For example,

```
EvAX.exe    parameters.dat -md input.xyz
```

will calculate averaged EXAFS spectrum for atomic configurations, saved in "*input.xyz*" file with parameters as specified in "*parameters.dat*" file.

Importantly, since the atomic structure is read from a file and contain no information about periodicity of the material, such calculations should always be carried out without periodic boundary conditions, i.e., **pbc** parameter in "*parameters.dat*" file should be always set to 0.

If one, nevertheless, needs to calculate spectrum corresponding to infinite crystal, one can still do it, assuming that all atoms in such crystal are equivalent. Thus one can calculate an averaged spectrum only for one atom that is located sufficiently far away from the surface of the cluster, defined in "*input.xyz*" file. For this purpose the label of this particular atom in XYZ file needs to be changed (e.g., from "Pt" to "Pt0"), and the value of **Edge** parameter in "*parameters.dat*" file should also be changed accordingly (to "Pt0" in this example). If XYZ file contains many atomic configurations, the label change should be done only for the first configuration (labels for the subsequent configurations do not affect EvAX calculations. Note that EvAX assumes that the order of atoms in all configurations is the same). EvAX can also find such central absorbing atom automatically. For this purpose use option **-setcenter**:

```
EvAX.exe    parameters.dat -setcenter -md input.xyz
```

In the "*parameters.dat*" file, **Edge** parameter in this case should still be manually changed to, e.g., "Pt0", while in the "*input.xyz*" file atoms should have type, e.g., "Pt". EvAX then will find in the "*input.xyz*" the central atom of type "Pt", and change its name to "Pt0".

The averaged spectra will be saved with the name "*averaged_EXAFS_n*", where n is the number of atomic configurations, used to calculate the averaged spectrum. New "*averaged_EXAFS_...*" file will be generated after reading each **time2** atomic structures from XYZ file.

5. Comparison of EXAFS data

5.1 *k*,*q* and *R* spaces

Comparison of two EXAFS spectra needs to be carried out at least once at each EvAX iteration: we compare the total configuration-averaged theoretical EXAFS spectrum $\chi_{\mathrm{tot}}(k)$ that corresponds to the current atomic configuration with the experimental spectrum $\chi_{\mathrm{exp}}(k)$.

Such a comparison can be carried out directly in *k*-space. Alternatively, one can perform the well-known Fourier transform (FT) that provides the representation of the signal in the frequency domain, and perform comparison of signals in the frequency space. Note that in the case of EXAFS analysis the frequencies of signal correspond to interatomic distances *R*. This approach allows one not only to compare spectra, but also to filter out both the low frequency (associated mainly with the problems of background substraction) and high frequency noise from experimental spectra, and to exclude from the experimental spectra the contributions of the further coordination shells, by choosing for comparison the appropriate range in *R*-space. One also can carry out inverse Fourier transform, and get back to the representation of spectra in *k*-space, thus essentially performing Fourier filtering. Comparison of Fourier-filtered spectra is sometimes referred to as comparison in *q*-space. Even better approach often is to replace Fourier transform by so called wavelet transform (WT). WT can be more informative than the FT, since it provides 2D-representation of our signal, revealing the features of signal both in *k* and *R*-spaces simultaneously. WT is closely related to a so called Short Time Fourier Transform (STFT), also known as Gabor transform.

To select, which approach should be used for spectra comparison, use parameter **Space**, which can have values **k**,**R**, **q**, or **w**, corresponding to the comparison in *k*, *R*, *q* or wavelet spaces. For example,

```
Space    w
```

indicates that wavelet space will be used for comparison of the experimental and theoretical EXAFS data during EvAX simulations (this option is used by default).

5.2 Euclidean distance

To measure the difference between two EXAFS spectra in k-space, we can introduce the quantity ξ_k that is defined as

$$\xi_k = \frac{\|\chi_{\text{tot}}(k)k^n - \chi_{\text{exp}}(k)k^n\|_\rho}{\|\chi_{\text{exp}}(k)k^n\|_\rho}. \tag{5.1}$$

Thus during the EvAX simulations we minimize the difference between the theoretical and experimental EXAFS spectra in k-space. The vertical lines $\|...\|_\rho$ denote here the Euclidean norm with optional weight function ρ, and k^n (n=1, 2, or 3) is the weighting factor, commonly used in EXAFS analysis to compensate the damping of spectrum at high k-values.

Formally, Euclidean norm of some function $f(k_1, k_2, ..., k_l)$ is defined as $\|f(k_1, k_2, ..., k_l)\|_\rho = \left(\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty}...\int\limits_{-\infty}^{\infty}|f(k_1, k_2, ..., k_l)|\rho((k_1, k_2, ..., k_l))dk_1dk_2...dk_l\right)^{1/2}$.

In reality, of course, the integration is replaced by summation: $\|f(k)\|_\rho = \left(\sum\limits_{i=0}^{N}|\tilde{f}(k_i)|\rho(k_i)\Delta k\right)^{1/2}$, where $\tilde{f}(k_i)$ is obtained as an interpolation of the experimental or theoretically calculated spectra $f(k)$, which then is discretized on a mesh in $k-$space with step Δk.

Weight function $\rho(k)$ normally is just a rectangular window function, which is equal to 1 for $k_{\text{min}} < k < k_{\text{max}}$ and is 0 otherwise.

Parameters n (power of k in the weighting factor), Δk, k_{min} and k_{max} are provided as parameters **k_power**, **dk k_min** and **k_max**. For example,

```
Space     k
k_power   2
dk        0.05
k_min     3.0
k_max     15.0
```

specifies that comparison of theoretical and experimental EXAFS spectra will be carried out in k-space in the range between 3.0 and 15.0 Å^{-1} with k-step 0.05 Å^{-1}, and both EXAFS spectra will be multiplied by k^2 before the comparison.

If one wants to apply some different (non-rectangular) weights, there is a trick that can be used for this purpose. Note that before the first iteration, EvAX saves the tabulated $\rho(k)$ function to a text file "*window_K_0*" (if more than one experimental spectrum is used simultaneously, each may have different $\rho(k)$ function, and corresponding text files will be "*window_K_0*", "*window_K_1*", "*window_K_2*", ...).

This function can be edited manually, and then EvAX calculations can be restarted with an additional option **-skipwin**:

```
EvAX.exe   parameters.dat   -skipwin
```

-skipwin option prevents EvAX from creating and saving $\rho(k)$ function, and instead this function will be read from the existing "*window_K_0*" file in the folder with EvAX executable.

5.3 Fourier transform

Instead of using the full experimental EXAFS spectrum, one can also use as function $\chi_{\text{exp}}(k)$ the spectrum, filtered by Fourier transform in the specified R-space range. For our purposes we define Fourier transform of some EXAFS spectrum $\chi(k)$ as

$$\text{FT}_\chi(R) = \int\limits_{k_{\text{min}}}^{k_{\text{max}}} \left(\frac{\cos(2Rk)}{n_{\text{Re}}(R)} + i\frac{\sin(2Rk)}{n_{\text{Im}}(R)} \right) \chi(k)k^n g(k)dk \tag{5.2}$$

and inverse Fourier transform as

$$\text{BFT}_\chi(k) = \frac{\sigma_g}{g(k)\sqrt{\pi}} \int\limits_{R_{\text{min}}}^{R_{\text{max}}} e^{-2iRk}\text{FT}_\chi(\text{R})\rho_{\text{R}}(\text{R})\text{dR} \tag{5.3}$$

Here

$$n_{\text{Re}}(R) = \int\limits_{k_{\text{min}}}^{k_{\text{max}}} \cos^2(2Rk)g(k)dk, \tag{5.4}$$

$$n_{\text{Im}}(R) = \int\limits_{k_{\text{min}}}^{k_{\text{max}}} \sin^2(2Rk)g(k)dk, \tag{5.5}$$

and $g(k)$ is the window-function. By default Gaussian window function is used $g(k) = e^{-((k-k_0)/\sigma_g)^2}$ with $k_0 = (k_{\text{min}} + k_{\text{max}})/2$ and $\sigma_g = (k_{\text{max}} - k_{\text{min}})/(2\sqrt{\ln 10})$, which damps the spectrum at $k = k_{\text{min}}$ and $k = k_{\text{max}}$, and, hence, allows us to avoid the problems that arise due to the finite-length of the spectrum [14, p.111]. Such Fourier filtering allows us to remove the contributions of scattering paths longer than $2R_{\text{max}}$ and also the high-frequency experimental noise. Instead of Gaussian window function, one can also use "Hanning" window function with tapering parameter Δk. In the latter case, $g(k) = 1$ for k values between $k_{\text{min}} + \Delta k$ and $k_{\text{max}} - \Delta k$, is equal to $\cos^2(((k - k_{\text{min}})/\Delta k - 1)\pi/2)$ for k values between k_{min} and $k_{\text{min}} + \Delta k$. Moreover, $g(k)$ is equal to $\cos^2(((k_{\text{max}} - k)/\Delta k - 1)\pi/2)$ for k values between $k_{\text{max}} - \Delta k$ and k_{max}. If "Hanning" window is used, the integral in the $\text{FT}_\chi(R)$ expression is also multiplied by a constant $C = (k_{\text{max}} - k_{\text{min}} - \Delta k)/(2\sqrt{\pi})$. This convention makes the obtained Fourier transform result to be directly comparable with the output of codes for XAS data processing, such as ATHENA[1]. To choose whether Gaussian or "Hanning" window will be used, specify the **FT_window_type** parameter. If it is set to any negative number, then Gaussian window will be used. If it is set to any positive number Δk, then "Hanning" window will be used with the specified tapering parameter.

The difference between the experimental and simulated spectra now can be defined as

$$\xi_q = \frac{||\text{BFT}_{\text{tot}}(k) - \text{BFT}_{\text{exp}}(k)||_{\rho_q}}{||\text{BFT}_{\text{exp}}(k)||_{\rho_q}}. \tag{5.6}$$

[1] Note that ATHENA and EvAX use different convention for k_{min} and k_{max}. Fourier transform carried out by EvAX between k_{min} and k_{max} is equivalent to carrying out Fourier transform by ATHENA between $k_{\text{min}} + 0.5\Delta k$ and $k_{\text{max}} - 0.5\Delta k$

Instead of performing inverse Fourier transform, one can also use the Fourier images of the total theoretical and experimental EXAFS spectra directly, and carry out the minimization of difference between spectra in the frequency space (R-space). In this case the difference between both spectra is defined as

$$\xi_R = \frac{\|\mathrm{FT}_{\mathrm{tot}}(R) - \mathrm{FT}_{\mathrm{exp}}(R)\|_{\rho_R}}{\|\mathrm{FT}_{\mathrm{exp}}(R)\|_{\rho_R}}. \tag{5.7}$$

Weight functions ρ_R and ρ_k by default are, again, rectangular functions, which are equal to 1 in the ranges between R_{min} and R_{max}, and k_{min} and k_{max}, correspondingly. After the first EvAX iteration, these weight functions are saved to disk with filenames like "*window_R_0*" and "*window_Q_0*". Similar approach as described above for k-space can be used to replace them with something more sophisticated.

For example,

```
Space           r
k_power         2
k_min           3.0
k_max           15.0
R_min           1.0
R_max           5.0
FT_window_type  -1
```

specifies that comparison of theoretical and experimental EXAFS spectra will be carried out in R-space in the range between 1.0 and 5.0 Å, Fourier transform will be carried out in the k-range between 3.0 and 15.0 Å^{-1}, tapered with a Gaussian window function, and both EXAFS spectra will be multiplied by k^2 before the FT. To run simulations in q-space, just specify in the example above

```
Space   q
```

Note that according to Parseval's theorem,[2] ξ_k and ξ_R should be equal. In practice, however, since the signals used for the analysis are limited both in k- and R-spaces, the ξ_k and ξ_R quantities are slightly different, and can be used to emphasize discrepancies of different features of the compared spectra.

As a rule of thumb, the agreement between the experimental and theoretical spectra can be considered to be reasonably good, if ξ_k or ξ_R is smaller than 0.2 (see [14, p.304] and [15]). Note also that ξ factor, defined as above, is essentially equivalent to the square root of R-factor, reported by popular ARTEMIS/FEFFIT software forthe conventional EXAFS data fitting [7] .

5.4 Wavelet transform

Besides comparison of EXAFS spectra in either k-, q- or R-spaces, there is another possibility: we can compare EXAFS spectra in k- and R-spaces *simultaneously* using the so called wavelet transform. WT is a modern technique of spectrum processing, and during the last fifteen years the advantages of

[2] *Parseval's theorem* states that Euclidean L_2 norms of spectrum and its Fourier image are equal, i.e., $\|\chi(k)\| = \|\mathrm{FT}_\chi(\mathrm{R})\|$

its application to EXAFS analysis have been demonstrated [16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]: WT has been found useful for EXAFS data extraction, noise filtering and de-glitching procedures and qualitative EXAFS analysis.

Wavelet transform of a given EXAFS spectrum $\chi(k)$ is defined as [19]

$$w(R,k) = \sqrt{R/R_0} \int_{-\infty}^{+\infty} \chi\left(k'\right) k'^n \varphi\left(\left(R/R_0\right)\left(k'-k\right)\right) dk'. \tag{5.8}$$

In fact, it is simply the inner product of the analyzed spectrum and translated by k-units and R/R_0-times distorted function $\varphi(k)$ (this function $\varphi(k)$ is called "mother-wavelet"). The basic idea behind the wavelet analysis is the Cauchy − Bunyakovsky − Schwarz (CBS) inequality that states that the inner product of two functions with fixed norms is the largest, when these functions have similar shapes.[3] The aim of wavelet transform is to decompose the analyzed spectrum in k and R-spaces, i.e., to reveal, which frequencies (R-values) are present in the spectrum, and, simultaneously, at which values of wavenumber k the components, corresponding to each R-value, do contribute. If as a mother-wavelet we choose such function $\varphi(k)$ that is (i) well localized in k-space and (ii) well-localized in R-space (i.e., it has well defined characteristic frequency R_0), the interpretation of the obtained wavelet image is rather straightforward: the absolute value of wavelet image of spectrum $\chi(k)$ at some point (k_1, R_1) is large, if in this spectrum around wavenumber point $k = k_1$ the amplitude of the spectral component that corresponds to frequency $R = R_1$ is significant. Note that if we neglect the first condition and mother-wavelet is localized only in R-space, the wavelet transform gets actually equivalent to Fourier transform.

One of the possible mother-wavelet functions that have the properties, mentioned above, and that are suitable for EXAFS analysis, is the so-called Morlet wavelet [19]: for Morlet WT, function $\varphi(k)$ is defined as

$$\varphi_{\text{Morlet}}(k) = \exp\left(-2iR_0 k\right) \exp\left(-\sigma_0^2 k^2\right). \tag{5.9}$$

For illustration, let us apply Morlet WT to a simple model spectrum:

$$\chi(k) = \sum_{i=1}^{3} \sin(2R_i k) e^{-2\sigma^2 (k-k_i)^2} \tag{5.10}$$

with $R_1 = 2$ Å, $R_2 = 6$ Å, $R_3 = 10$ Å, $\sigma^2 = 0.1$ Å^2, $k_1 = 9$ Å^{-1}, $k_2 = 15$ Å^{-1}, $k_3 = 21$ Å^{-1}. This spectrum is plotted in the upper part of Fig. 5.4, its Fourier transform - in the right panel of Fig. 5.4. The modulus of Morlet WT is shown in the central panel of Fig. 5.4. While in the case of Fourier transform any information on positions of the different components of model spectrum in k-space (i.e., the values of k_1, k_2, k_3) is lost, wavelet transform gives us the two-dimensional representation of the studied spectrum and provides information on localization of its components both in $R-$ and $k-$spaces. This is an important advantage for the analysis of EXAFS data from samples with different types of atoms: it is known that heavy atoms usually contribute to the total EXAFS spectrum at higher values of wavenumber k than light atoms, thus the contributions of atoms of different type can be separated using wavelet analysis.

Fig. 5.4 reveals also one of the peculiarities of wavelet transform. The studied model spectrum consists of three components, and duration of each component in $k-$space is the same. Therefore

[3] More precisely, CBS inequality states that for the inner product of functions g and h, denoted as (g,h), the following relation holds: $|(g,h)| \leq \|g\|\|h\|$, and the equality is achieved only if functions g and h are linearly dependent

all three peaks in the Fourier image of this spectrum are also similar. Three peaks in the wavelet image, in turn, have different shapes: peaks that correspond to higher frequencies are distorted in the direction of $R-$axis. This is a result of general property of the wavelet transform - it has different resolution at different frequencies. The image of low-frequency components (components with frequency R smaller than characteristic frequency of the mother-wavelet R_0) will have high resolution in $R-$space and poor resolution in $k-$space, while the image of high-frequency components (with R larger than R_0) will be well resolved in $k-$space, but in R-space their resolution will be worse [18].

To be more specific, let us look at Morlet WT of damped harmonic functions

$$\chi(k) = \exp(2iR_1 k)\exp\left(-\sigma_1^2 (k-k_1)^2\right) \tag{5.11}$$

We insert this expression into Eq. (5.8) and after integration we obtain

$$|w(R,k)| = \sqrt{\frac{\pi R R_0}{\sigma_0^2 R^2 + \sigma_1^2 R_0^2}} e^{-\frac{R_0^2 (R-R_1)^2}{\sigma_0^2 R^2 + \sigma_1^2 R_0^2}} e^{-\frac{R^2 \sigma_0^2 \sigma_1^2 (k-k_1)^2}{\sigma_0^2 R^2 + \sigma_1^2 R_0^2}} . \tag{5.12}$$

The first exponent provides the localization of wavelet image in $R-$space: it is Gaussian function centered at R_1 and with characteristic width $\Delta_R = \sqrt{\sigma_0^2 R^2 + \sigma_1^2 R_0^2}/R_0$. The second exponent, in turn, provides the localization of wavelet image in $k-$space: again it is Gaussian function now centered at k_1 and with characteristic width $\Delta_k = \sqrt{\sigma_0^2 R^2 + \sigma_1^2 R_0^2}/(R\sigma_0\sigma_1)$. The distortion of the peak of wavelet image can now be characterized by ratio $\Delta_R/\Delta_k = \sigma_0 \sigma R/R_0$. And again, one can see, the larger is the frequency R, the more distorted is the wavelet image in the direction of R-axis.

Note also, for given value of R the Δ_k is the larger, the larger is the parameter R_0, but Δ_R is inversely proportional to R_0. It means, we can improve the resolution in k-space by reducing the value of parameter R_0; however, this will result in poorer resolution in R-space, and vice versa: the resolution in R-space can be increased by reducing the value of R_0, but, consequently, the resolution of WT in $k-$space will be poorer in this case (see Fig. 5.4). We can see thus, wavelet transform can be a flexible tool for the spectrum analysis and it can easily be adapted to emphasize the spectrum features of interest [19].

From the practical point of view, wavelet transform has one useful property that allows an efficient implementation of WT algorithm. Note that Eq. (5.8) that defines the wavelet transform of function $\chi(k)$, looks similar to the convolution of functions $\chi(k)$ and $\phi(k)$. Hence the result, similar to the well known convolution theorem can be obtained: the Fourier transform of the wavelet image can be written as

$$\begin{aligned}
\mathrm{FT}_{w(R,k)}(\omega) &= \sqrt{R/(2\pi R_0)} \int_{-\infty}^{+\infty} dk e^{i\omega k} \int_{-\infty}^{+\infty} \chi\left(k'\right) \varphi\left((R/R_0)\left(k'-k\right)\right) dk' \\
&= \sqrt{R/(2\pi R_0)} \int_{-\infty}^{+\infty} dk \int_{-\infty}^{+\infty} e^{i\omega k'} \chi\left(k'\right) e^{-i\omega x (R_0/R)} \varphi(x)(R_0/R) dx \\
&= \sqrt{R_0/(2\pi R)} \int_{-\infty}^{+\infty} dk e^{i\omega k'} \chi\left(k'\right) \int_{-\infty}^{+\infty} dx e^{-i\omega x (R_0/R)} \varphi(x) dx \\
&= \sqrt{2\pi R_0/R}\, \mathrm{FT}_{\chi}(\omega) \mathrm{FT}_{\chi}(-\omega R_0/R),
\end{aligned} \tag{5.13}$$

and thus one gets

$$w(R,k) = \sqrt{2\pi R_0/R}\, \mathrm{BFT}\left[\mathrm{FT}_{\chi}(\omega) \mathrm{FT}_{\varphi}(-\omega R_0/R)\right] . \tag{5.14}$$

The Fourier transform of mother-wavelet function can be calculated analytically. In the case of Morlet wavelet transform with $\varphi(k) = e^{-2iR_0k}e^{-\sigma_0^2k^2}$, the corresponding result is simply Gaussian function: $\mathrm{FT}_\varphi(\omega) = \frac{1}{\sqrt{2}\sigma_0}e^{-(\omega-2R_0)^2/(4\sigma_0^2)}$. The Fourier transform of the analyzed data $\chi(k)$ and inverse Fourier transform, in turn, can be carried out using efficient fast Fourier transform algorithm (Cooley-Tukey algorithm). Besides allowing fast calculations of wavelet transform, Eq. (5.14) provides also an useful interpretation of wavelet transform: one can see, Morlet wavelet transform can be considered as Fourier filtering of the analyzed spectrum with the Gaussian filter with variable width.

Other properties of wavelet transform in details are described in, for example, [27] and [28].

For the comparison of EXAFS spectra in the RMC and EA simulations, WT can be employed similarly as Fourier transform. The difference between spectra now can be calculated as

$$\xi_{k,R} = \frac{\|w_{\mathrm{tot}}(R,k) - w_{\mathrm{exp}}(R,k)\|_{\rho_w}}{\|w_{\mathrm{exp}}(R,k)\|_{\rho_w}}, \tag{5.15}$$

where w_{tot} and w_{exp} are the Morlet wavelet transforms of calculated and experimental EXAFS spectra ($\chi_{\mathrm{tot}}(k)k^n$ and $\chi_{\mathrm{exp}}(k)k^n$), respectively.

To use wavelet transform for comparison of theoretical and experimental EXAFS data, use

```
Space    w
```

(it is a default value for **Space** parameter).

For example,

```
Space        w
k_power      2
k_min        3.0
k_max        15.0
R_min        1.0
R_max        5.0
R0_for_WT    3.0
```

specifies that comparison of theoretical and experimental EXAFS spectra will be carried out in wavelet-space in the R-range between 1.0 and 5.0 Å, Fourier transform will be carried out in the k-range between 3.0 and 15.0 Å^{-1}, and both EXAFS spectra will be multiplied by k^2 before the WT.

Parameter **R0_for_WT** specifies the characteristic frequency R_0 of Morlet wavelet. If it is not specified or is set to a negative value, a default value equal to an average of **R_min** and **R_max** values will be used. σ_0 parameter that specifies the localization of mother wavelet in k-space is always equal to 1 Å (note that effect of σ_0 parameter on WT is very similar to that of R_0 parameter).

Thus wavelet transform is an unambiguous linear transformation of signal that reflects its frequency content. In that sense it is fully analogical to the Fourier transform, widely used in the EXAFS community. A good agreement between two spectra in the wavelet space automatically means a good agreement between them in k-space and R-space. The opposite, dealing with signals, limited in k- and R- spaces, is not always true: a good agreement, for instance, in Fourier space only, does not necessarily mean a good agreement in k-space, and, consequently, also in wavelet space. Thus WT is a natural way to take into account features of signal both in k- and Fourier space simultaneously. Additionally, the advantage of WT over Fourier transform and over the representation of spectra in k-space can be seen in the case when the agreement between two spectra is *not good*: in this situation WT, providing the information on the misfit of spectra both in k- and frequency space simultaneously, allows one to identify the problem more easily.

Therefore we believe that the WT-based criterion for the comparison of theoretical and experimental data (Eq. (5.15)) will provide the best results in most EvAX simulations.

The additional advantage that is provided by the use of wavelet transform, is the possibility to change the sensitivity to the different components of spectra, using weight function $\rho_w(R,k)$. Again, by default, as $\rho_w(R,k)$ a rectangular function, which is equal to 1 in the range between $R_{\min}$ and $R_{\max}$, and $k_{\min}$ and $k_{\max}$. After the first EvAX iteration, this weight functions are saved to disk with filenames like "*window_W_0*". Similar approach and **-skipwin** option as described above can be used to replace it with something else to increase importance of specific regions in (k,R) space.

5.5 Short time Fourier transform (Gabor transform)

If in Eq. (5.8) we change the value of WT parameter R_0 for different interatomic distances R and keep $R_0 = R$, Morlet WT becomes equivalent to so called Short time Fourier transform, or Gabor transform. Since the resolution of STFT does not depend on frequency of analyzed signal, the interpretation of STFT features can sometimes be more straightforward than that of Morlet WT, especially for signals with narrow frequency range. To use STFT instead of WT, set parameter **WT_type** to 0. The parameter **R0_for_WT** then will have no effect on calculations.

5.6 FT, WT or STFT for a given spectrum

Note that one can use EvAX to calculate Fourier, inverse Fourier and wavelet transforms for given spectrum without running RMC simulations. For this purpose one needs to run EvAX with additional keyword **-ftwt**. For example,

```
EvAX.exe    parameters.dat -ftwt
```

will calculate Fourier, inverse Fourier and wavelet transforms for the spectrum with the name specified as **File_with_experimental_EXAFS_signal** and with the FT and WT parameters as specified in "*parameters.dat*" file. Alternatively, the name of file with EXAFS data can by supplied via command line:

```
EvAX.exe    parameters.dat -ftwt -spectrum EXAFS.dat
```

5.7 Residual in calculations with multiple absorption edges

When simultaneously fitting several EXAFS spectra, EvAX minimizes the total residual, defined as

$$\xi_{TOT} = \sum w_i \xi_i. \tag{5.16}$$

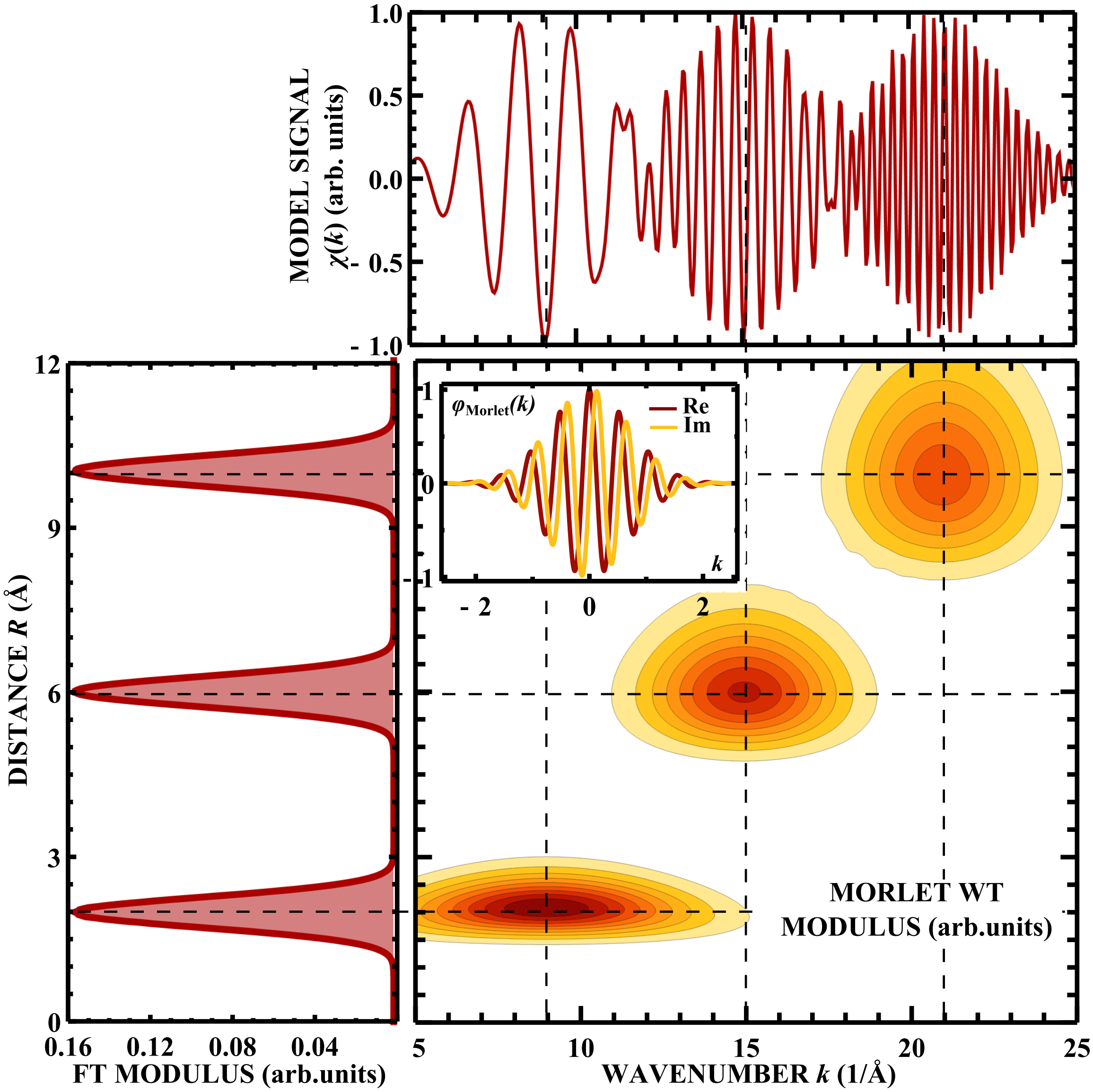

Figure 5.1: Morlet wavelet transform in action
Model spectrum (*upper panel*), modulus of its Fourier transform (*left panel*) and modulus of its Morlet wavelet transform (*central panel*). In the *inset* - real and imaginary parts of Morlet mother-wavelet.

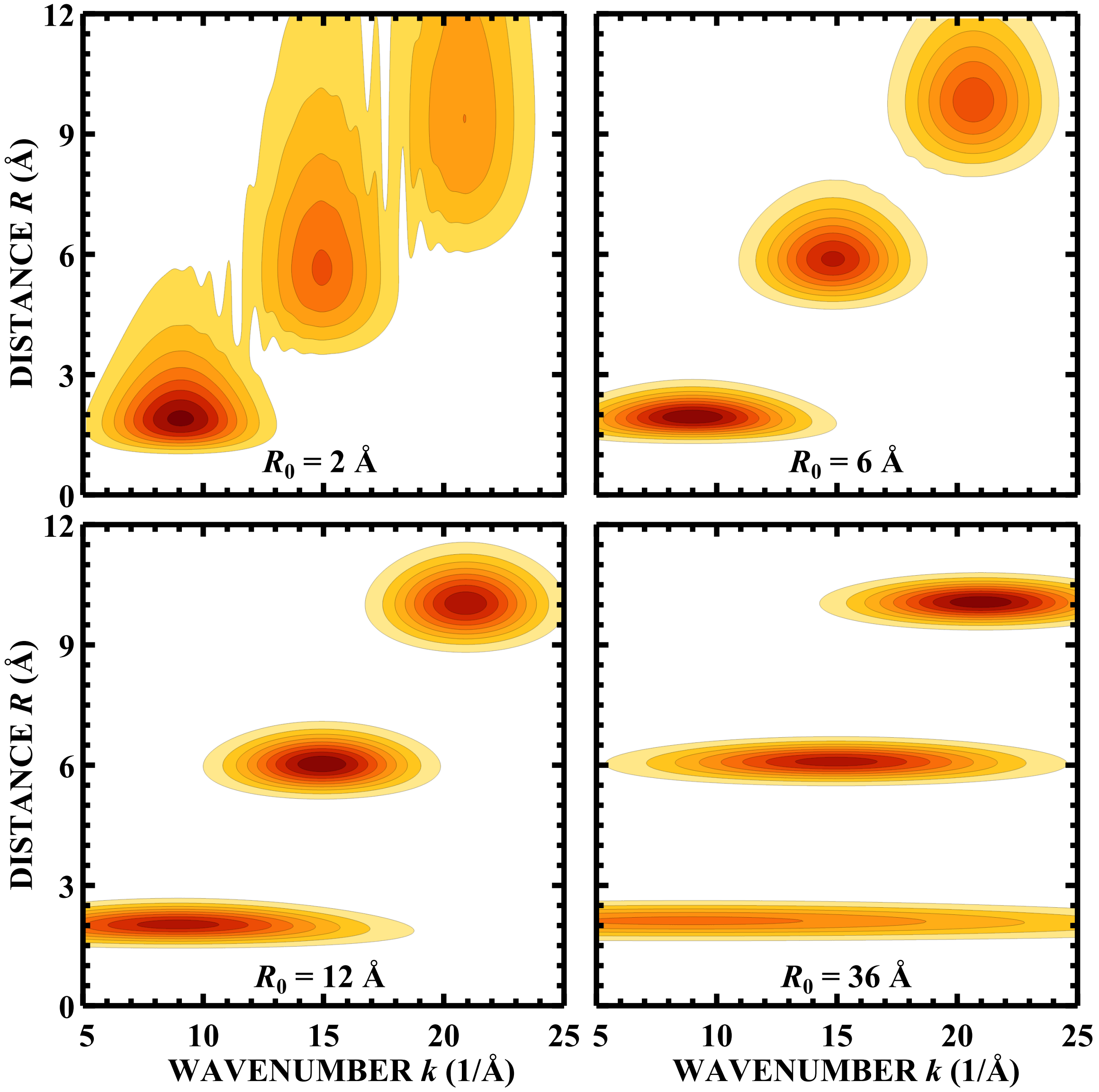

Figure 5.2: Controlling the resolution of Morlet wavelet transform. Modulus of the Morlet wavelet transform of model spectrum, calculated with different characteristic frequency values R_0 for mother-wavelet.

Here ξ_i is the residual for the i-th spectrum, and w_i is its weight. The values of weights w_i can be specified explicitly by using parameter **Spectrum_weight**. For example, if one spectra has lower quality than the other, its weight can be decreased. If **Spectrum_weight** is set to a negative value, ξ_{TOT} will be just an average of ξ_i values.

It is important to note that the weights should reflect also how strongly the same structural changes affect the residual for different absorption edges. For a material where the concentrations of absorbers of different types differ strongly, small displacement of all atoms can have a minor impact on the majority species, but a much stronger effect on the residual for a minority species. In this case, if weights for both spectra are the same, EvAX might focus too strongly on improving the residual for the minority species, neglecting a large disagreement between experiment and simulations for the majority species. It is therefore recommended to allow EvAX to adjust weights w_i automatically by setting parameter **adjust_spectra_weights** to 1. In this case weights w_i will be set to be inversely proportional to the average change in the residual for each EXAFS spectrum.

For example,

```
Number_of_spectra        2
Spectrum_weight          0.5   0.5
adjust_spectra_weights   1
```

will indicate that in the calculations where two spectra are fitted simultaneously, the starting weights for the residuals for these two spectra are equal to 0.5, but these will be adjusted during the simulations.

6. Structure model and its optimization

6.1 Initial structure model

In the case of crystalline systems the equilibrium structure is usually known: the lattice parameters of crystalline material can be determined by diffraction techniques with much higher accuracy (better than 10^{-3} Å), compared to that provided by modern EXAFS analysis (usually about 10^{-2} Å). Therefore in our approach the RMC simulation of EXAFS spectrum is performed using a fixed supercell size, defined by the lattice parameters, and by initial placing of atoms at proper Wyckoff positions. This allows us to account for the information available from diffraction data without direct simulations of diffraction pattern. However, small random initial displacements for all atoms can be given to include approximately thermal disorder and, thus, to avoid the rapid changes of residual at the beginning of RMC iteration process. The shape of the supercell is determined by the crystal symmetry and is not necessarily cubic.

EXAFS method is sensitive to the local atomic structure (usually up to 10 Å around the absorbing atom) due to the restrictions imposed by the life time of the excitation and thermal disorder. Therefore, in a periodic system one can probe and needs to account for a relatively small amount of atoms in a rather small cell, whose size should be at least twice the largest radial distance in the Fourier transform of EXAFS spectrum. This means that for the simulations of EXAFS spectra from crystalline materials a supercell containing a few hundreds of atoms should be sufficient. To compare, the number of atoms, required for the RMC modelling of disordered materials, is at least 1000 [29].

Following parameters are used to define the geometry for EvAX simulations. **File_with_structure** parameter specifies the name of text file that describes the geometry of unit cell. The possible formats of this file are described in the next Section For example,

```
File_with_structure    input_files/structure.p1
```

will try to read the structure from "*structure.p1*" file in "input_files" directory. If the file does not exist, EvAX will print out error message.

It is also possible to re-scale the atomic coordinates used by EvAX without modifying the structure file. For this purpose **Scale_coordinates_by** parameter can be used. For example,

```
Scale_coordinates_by    1.05
```

will isotropically re-scale the structure by 5%.

Parameter **Supercell** sets the size of simulation box (in unit cells). For example,

```
Supercell    4x4x4
```

constructs a supercell consisting of 4 × 4 × 4 unit cell.

Syntax used in older EvAX versions is also accepted:

```
Supercell    4 4 4
```

If the first number in the supercell parameter is larger than 0, but the second or the third parameter is negative, EvAX will interpret the first number as the desired total number of atoms in the supercell, and will choose the supercell size so that it has approximately this specified number of atoms, and the shape of the cell is as close as possible to a cubic one. For example,

```
Supercell    100x-1x-1
```

will create a supercell with approximatelly 100 atoms in total.

Finally, **pbc** flag specifies, if periodic boundary conditions will be employed. Use

```
pbc    1
```

for crystalline materials and

```
pbc    0
```

for non-periodic materials.

Note that while the terms "equilibrium structure" and "lattice constants" cannot be directly applied to describe small nanoparticles and other non-periodic objects, where surface effects play an important role, a supercell concept is useful in this case as well. Note that the experimental EXAFS signal, e.g., of metallic nanoparticles normally is an average of signals, produced by many separate nanoparticles. Therefore, for more accurate description of the material, it is more appropriate in this case to have a structure model also consisting of several separate nanoparticles, rather just a single nanoparticle. To generate the initial configuration for such model, one can define the positions of atoms within one nanoparticle (similarly as one defines the positions of atoms within a crystalline unit cell), and then multiply this structure model, by translating the original nanoparticle along x, y and z directions, similarly as crystalline unit cells are translated to construct a supercell, representing structure of crystalline material. The "lattice constants" in the case of nanoparticles can be arbitrary numbers, but large enough to keep different replicas of the original particle at sufficient distances from each other.

6.2 Structure file

The lattice parameters and types of atoms in the structure are specified in a separate text file. The name of this file should be specified in the parameter file (parameter **File_with_structure**).

EvAX recognizes three different types of structure input files (the different formats are recognized based on the file name extension):

6.2.1 CIF files

File_with_structure **input_files/structure.cif**

For periodic crystal structures the easiest is to use directly CIF file that can be downloaded, e.g., from databases such as American Mineralogist Crystal Structure Database (http://rruff.geo.arizona.edu/AMS/amcsd.php), Materials Proejct (https://next-gen.materialsproject.org/) or many others. Note that while this approach typically works well, the formatting of some CIF files cannot be reckognized properly by EvAX. In these cases, the safest option is to use P1 file for structure definition, as explained below.

6.2.2 XYZ files

File_with_structure **input_files/structure.xyz**

For non-periodic structures, the most convenient is to use standard XYZ chemical file format is used here: first line contains total number of atoms, second line is comment line, and the remaining rows in the file each corresponds to one atom, and contains its label, and *x*, *y* and *z* coordinates.

6.2.3 P1 files

File_with_structure **input_files/structure.p1**

P1-type file can be created manually, or with appropriate software (for example, freely available program VESTA can be used to create the structure or import it (e.g., from CIF-type file), and then export it as **.p1*-type file [6]. P1 files can be used both for periodic and non-periodic structures. In fact, when importing CIF or XYZ type files, EvAX first converts them into P1 file, saved as *temp_P1.p1*, so that user can verify that the structure has been interpreted properly.

An example of such file is given below:

```
PtO2
1.0
        4.4879999161         0.0000000000         0.0000000000
        0.0000000000         4.5329999924         0.0000000000
        0.0000000000         0.0000000000         3.1380000114
   Pt     O
    2     4
Direct
     0.000000000         0.000000000         0.000000000  Pt
     0.500000000         0.500000000         0.500000000  Pt
     0.280999988         0.347999990         0.000000000  O
     0.718999982         0.652000010         0.000000000  O
     0.219000012         0.847999990         0.500000000  O
     0.781000018         0.152000010         0.500000000  O
```

The structure of this file is strict: the order of parameters in the file is important, and also no additional lines can be added. File starts with some arbitrary title, then follows the description of the lattice vectors in form of n_x, n_y, n_z, where n_x, n_y, n_z are the components of the lattice vectors. Further follows two rows that are actually not used by EvAX. VESTA program prints here the list of atoms that are present in the structure, and their total number in the list below, but user can treat these two lines as additional comment lines.

Next line describes the type of used coordinate systems. It can be either `Direct` or `Cartesian`. The former indicates that fractional coordinates will be used. The latter indicates that coordinates below are Cartesian coordinates (these may be more convenient to set the structure of non-periodic models).

What follows next is the list of fractional or Cartesian coordinates of all atoms, where entry for each atom ends with its label. As a label one can use the chemical symbol of atom (Zn, O, etc.) or its chemical symbol with one additional digit appended (Zn1, O2, etc.): the atomic label will be used by EvAX program to identify atomic numbers for FEFF calculations. Additional information about each atom can be printed out in **.p1* file after atomic label. This information will be ignored by EvAX.

6.2.4 Atom labels

Regardless of file format used, EvAX accepts as atom labels either their chemical symbol (e.g., "Cu"), or chemical symbol with one digit added (e.g., "Cu0", "Cu1", etc.) By default, atoms with different labels are treated as non-equivalent species. For example, if one species as an absorbing species "Cu1", only atoms labelled with this label will be considered as absorbers. If instead one wants to neglect the numerical symbols and treat all species with the same chemical symbol but different digits in the label as equivalent, this can be specified by **labels_to_element_name** flag. For example,

labels_to_element_name	**1**

will force EvAX to discard the digits after the chemical symbol in the atom label, and to treat "Cu0",

"Cu1", etc as equivalent species.

6.3 Structure models with random atom replacements

For RMC simulations of random alloys, solid solutions or crystalline structures doped with other elements, EvAX provides a possibility to create the structure models where the randomly chosen atoms of one type are replaced by atoms of other type. To replace atoms of type A with atoms of type B, use **Doping** **A>B**. To specify, what fraction of A atoms needs to be replaced, use **Doping_level** parameter. For example,

Doping	**Pd>Ni**
Doping_level	**0.5**

will randomly replace 50% of the Pd atoms in the structure model with Ni atoms.

The default values for these parameters are **none** for **Doping** and -1 for **Doping_level**, both indicating that no replacements of atoms will take place.

6.4 Initial atomic displacements

Since even the crystalline material is locally always distorted, the equilibrium structure, defined by the initial structure file, as described above, usually gives quite bad agreement with experimental data. To save some calculation time, it is possible to apply some small random initial displacements for all atoms in the supercell, using **Initial_displacement** parameter. For example,

```
Initial_displacement    0.1
```

will displace all atoms in the structure model by vector $\vec{u} = \frac{u_{\text{init}}}{\sqrt{2}}(\rho_x, \rho_y, \rho_z)$, where ρ_x, ρ_y and ρ_z are random numbers between -1.0 and 1.0, sampled from uniform distribution, and u_{init} is **Initial_displacement** parameter value (0.1 Å in this example). Different sets of random numbers are generated for each of the atoms in the model.

Note that it is always better to have too small rather than too large value of **Initial_displacement**, since for RMC-type methods it is much easier to distort the system than to make it back ordered. The values close to the default value 0.1 Å typically work well.

It is also possible to estimate the optimal **Initial_displacement** using automated procedure. For this purpose, set **Initial_displacement** to any negative value (e.g., -1). In this case, before the main run, EvAX will try different values for u_{init} between 0.1 and 0.5 Å, and will compare the obtained agreements between experimental and simulated EXAFS spectra for the correspondingly distorted structures after the first iteration. Calculations will be continued with the u_{init} value that gave the best agreement.

Note that if u_{init} value, obtained in such automatic procedure, exceeds significantly the recommended value 0.1 Å, it signals that the chosen initial structure model is poorly describing the experimental data, and it can mean that something is wrong either with the structure model, or with the EXAFS data.

6.5 Atomic displacements for structure optimization

Once the initial atomic configuration is chosen, one should define the procedure for its modification using random atom displacements. For this purpose at each RMC step one can either randomly pick one atom and randomly change its coordinates, or can randomly modify coordinates of all atoms. EvAX supports both approaches, but the latter approach has been tested much more extensively and, hence, it is the default and recommended mode. To specify, which approach will be used during simulations, use **Move_type** parameter.

```
Move_type    move_all
```

will ensure that all atoms are displaced, while

```
Move_type    move_one
```

will ensure that only one, randomly selected atom is displaced. In both cases the atoms are displaced by vector $\vec{u} = u_{\max}(\rho_x, \rho_y, \rho_z)$, where ρ_x, ρ_y and ρ_z are random numbers between -1.0 and 1.0, sampled from uniform distribution, and $u_{\max}$ is the maximal allowed step size. The value for $u_{\max}$ is specified as **Maximal_step_length** parameter. For example,

```
Maximal_step_length    0.005
```

sets $u_{\max}$ to be equal to default value 0.005 Å. For calculations, where all atoms are displaced at each iteration, the recommended value for **Maximal_step_length** is between 0.001 and 0.01 Å. Too small value for this parameter obviously will result in a very slow initial evolution of the structure model. However, smaller step sizes typically result in lower final residual.

For calculations, where only one randomly selected atom is displaced, larger **Maximal_step_length** values can be used.

6.6 Constraints

Properly chosen constraints for atomic displacements are crucial for successful RMC simulations, since they reduce significantly the size of configurational space that needs to be explored, hence reduce the required computation time. They also allow one to reduce the ambiguity in the obtained structure model.

In the real crystals the displacements of atoms from their equilibrium positions due to thermal vibrations are normally less than a few tenths of angstrom. Therefore, in the EvAX code we usually constrain the displacements of atoms from their initial positions that are known from diffraction experiments to be smaller than some given value $\delta_{\max}$. In this case atoms are "boxed" in a sphere with radius $\delta_{\max}$. To apply this constrain, set the value of parameter **Potential_type** to **box**, and specify the $\delta_{\max}$ value using parameter **Maximal_displacement**. For example,

```
Potential_type         box
Maximal_displacement   0.4
```

are the default values for these parameters, and they confine atoms to the spheres with radii $\delta_{\max} = 0.4$ Å from their initial positions. Such displacements are usually sufficient to describe the disorder in crystalline materials. For more distorted systems, e.g., nanocrsytalline samples, one may need to use larger values of $\delta_{\max}$.

Instead of constraining the displacements of atoms from their initial positions, one can also constrain the minimal allowed interatomic distances. For this purpose set **Potential_type:** to **hardsphere**. The value, specified as **Maximal_displacement** in this case will be interpreted as the minimal allowed interatomic distance. For example,

```
Potential_type         hardsphere
Maximal_displacement   1.5
```

will prohibit the atomic configurations with interatomic distances shorter than 1.5 Å.

One can also use the combination of both constraints, and specify different constraints for atoms of different types. For such advanced control over constraints, set the value of **Potential_type** to **file**, and describe all constraints in a text file, as explained below. The name of this text file should be provided as parameter **File_with_potential**. For example,

```
Potential_type        file
File_with_potential   constraints.dat
```

will try to open file "*constraints.dat*" for information on constraints that should be used. Note that the value specified as **Maximal_displacement** will be ignored in this case, similarly as the value for **File_with_potential** will be ignored, if **Potential_type** is not set to **file**.

An example of file with constraints is given below:

```
BEGIN
box
Ge 0.4
hardsphere
Ge Ge 0.5
END
```

Similarly as parameter file, this file also starts with `BEGIN` keyword, and ends with `END` keyword. Between them in an arbitrary order user can specify the constraints that should be applied. For each constraint, one needs to specify its type (either `box` or `hardsphere`), and then the types of affected atoms. For `box`-type constraints only one type of atoms needs to specified. for `hardsphere`-type constraints two types of atoms need to be specified, since it is a pairwise constraint. Finally, a single number sets the maximal allowed atomic displacement from their initial position (for `box`-type constraint) or the minimal allowed interatomic distance (for `hardsphere`-type constraint). In the example above Ge atoms can displace no further than 0.4 Å from their initial positions, and, in addition, they cannot come at distances closer than 0.5 Å.

If some of the constraints are violated during the proposed atomic displacement, EvAX tries to find such structure model that is between the proposed and previous structure modules and that does not violate the constraints. Let us denote the $3N$-dimensional vector that characterizes the positions of all N atoms in the previous atomic configuration as $\vec{R}_{\text{old}}$, while the vector that describes the positions of these atoms after proposed atomic displacement as $\vec{R}_{\text{new}}$. Note that the structure corresponding to $\vec{R}_{\text{old}}$ always satisfies all constraints. Now, if the structure model that corresponds to vector $\vec{R}_{\text{new}}$ violates some constraints, we use instead for further calculations $\vec{R}'_{\text{new}} = 1/2\left(\vec{R}_{\text{old}} + \vec{R}_{\text{new}}\right)$. If $\vec{R}'_{\text{new}}$ also violates the constraints, we use instead $\vec{R}''_{\text{new}} = 1/2\left(\vec{R}_{\text{old}} + \vec{R}'_{\text{new}}\right)$, or continue this process till we find the structure model, where constraints are not violated.

EvAX code version 4 and older had a bug in the implementation of advanced constraints, which is fixed in EvAX-5 code.

6.7 Random number generation

The generation of pseudo-random numbers for atomic displacements and structure model optimization in our program is performed using the Mersenne-Twister algorithm [30]. The Mersenne-Twister algorithm is a common choice for many Monte Carlo implementations due to its speed, long repetition period and good statistical randomness of the generated numbers [31].

The sequence of pseudo-random numbers, used for EvAX calculations, is determined by the seed of random number generator. It consists of four integer numbers that are set by parameter **MT_seed**, for example,

MT_seed **123 456 7890 123**

If one wants to repeat calculations with different sequence of random numbers, one needs just to change any of the seed numbers. EvAX provides a convenient way to modify the seed parameter: by running

```
EvAX.exe    parameters.dat    -reseed
```

where "*parameters.dat*" is the name of the parameter file, one creates a copy of this file with the name "*parameters.dat_reseed*", where all calculation parameters are the same as in the original file, but the random generator seed is changed.

6.8 Metropolis algorithm

Here we discuss the algorithm that determines, whether the atomic configuration, proposed at some RMC iteration (i.e., obtained after atomic displacements), should be accepted or discarded. For this algorithm one needs to compare the theoretically calculated and experimental EXAFS data. In the previous Chapter different approaches for such comparison are discussed in details. At this point let us just assume that the quantity, characterizing the difference between two spectra, is known and is denoted by ξ.

Let the differences between total calculated and experimental EXAFS spectra for the current and new atomic configurations be equal to ξ^{old} and ξ^{new}, respectively. It is clear, if $\xi^{\text{new}} < \xi^{\text{old}}$, the proposed atomic configuration is better than the previous one and should be accepted. However, if we discard all atomic configurations, for which $\xi^{\text{new}} > \xi^{\text{old}}$, the difference ξ will always decrease and after some number of steps it will reach the *local* minimum. If the chosen initial structure model is close enough to the final solution, this may be already sufficient to obtain a good result. It is advised therefore always to check, whether this simple already allows to obtain a good agreement between experiment and simulations. To run the simulation where only the moves that improves the agreement between experiment and theory are accepted, use

```
adjust_acceptance    0
Acceptance_rate      -1
```

In other cases, in order to ensure that the *global* minimum is found, it is necessary to accept some of the atomic displacements for which $\xi^{\text{new}} > \xi^{\text{old}}$. Such strategy is realized in the most popular algorithm of the movement acceptance/discarding, proposed by Metropolis [32]

$$\begin{aligned}
&\text{if } \xi^{\text{new}} < \xi^{\text{old}}, \text{ the move is accepted;}\\
&\text{if } \xi^{\text{new}} > \xi^{\text{old}}, \text{ the move is accepted, if}\\
&\qquad\qquad \exp(-(\xi^{\text{new}} - \xi^{\text{old}})/\Theta) > r,\\
&\qquad\qquad \text{and discarded otherwise}
\end{aligned}\tag{6.1}$$

where r is a pseudo-random number in the range between 0 and 1, and Θ is a scaling parameter.

It can be shown, if one allows the atoms to move accordingly to Metropolis algorithm with fixed value of parameter Θ for a sufficiently long time, values of the ξ will fluctuate around its minimal value accordingly to Boltzmann distribution $P(\xi) \sim e^{-\xi/\Theta}$ [33].

The fact that the distribution of ξ is Boltzmann distribution means that at some arbitrary time moment the system most likely will be close to the state where ξ is close to 0. The standard deviation - the characteristic width of the Boltzmann distribution - is given by parameter Θ. Therefore, if Θ is large, the system will fluctuate around the global minimum with a large amplitude. On the other hand, if parameter Θ is small, we again will discard all the moves that increase the difference ξ, and the simulation will get stuck at some local minimum. The exact meaning of terms "large" and "small" here will depend on properties of the system: number of atoms in the model, step size for atomic displacements, derivatives of particular EXAFS spectra with respect to atomic coordinates. To avoid this ambiguity, it is more convenient to rewrite the last condition in Equation (6.1) as

$$p_0 \exp(\Delta\xi / \langle\Delta\xi\rangle) > r, \tag{6.2}$$

where $\Delta\xi = \xi^{\text{new}} - \xi^{\text{old}}$, $\langle\Delta\xi\rangle$ is the average change of the residual per iteration and p_0 is the acceptance rate. Note that the value of $\langle\Delta\xi\rangle$ depends on the system parameters (for example, it is smaller for systems with larger number of atoms).

In principle, as some other RMC codes, EvAX allows one to carry out RMC calculations with fixed value of p_0. For this purpose use, for example,

```
adjust_acceptance   0
Acceptance_rate     0.5
```

Here the first parameter specifies that the acceptance/discarding rate should be kept fixed during the simulations, while the second parameter sets the value for p_0).

The alternative and strongly recommended solution, implemented in our code, is to use so called simulated annealing approach. In this case the acceptance rate is adjusted during the simulations. For such calculations use

```
adjust_acceptance   1
Acceptance_rate     0.8
```

Specified **Acceptance_rate** value will then be treated as an initial value for the acceptance rate p_0.

6.9 Simulated annealing

In the simulated annealing approach [34], the parameter Θ and, hence, the acceptance ratio p_0 are not fixed but are decreased slowly. One starts with a large value of p_0 to stimulate a fast approach to the global minimum. Then the parameter p_0 decreases, so that the fluctuations of the system are damped. At the end of the simulation, p_0 is equal to 0 and, if the annealing has been carried out slowly enough, the system reaches the global minimum. The efficiency of this approach strongly depends on the so-called "cooling schedule" - the function that controls the decrease of acceptance probability during the simulation. Popular cooling schedules are, for instance, the so called Boltzmann annealing schedule, where $\Theta(t) \sim 1/\log(t)$, or the Cauchy annealing scheme, where $\Theta(t) \sim 1/t$. Many others are known and discussed, e.g., in [35]. For each particular problem one or another algorithm can be considered as the optimal, and also the optimal values of parameters for each schedule varies for different problems. Regardless of the cooling scheme we have chosen, one parameter always plays a crucial role: the length of the schedule $t_{\max}$, i.e., the number of iterations until the parameter Θ (or p_0) reaches zero. In fact, with any reasonable cooling scheme the global minimum can be found, if the length of the schedule is chosen long enough. Of course, since this parameter determines the length of our calculation, we are also interested to keep it as small as possible.

While some (complicated) theoretical results for special cases are available, in practice the prediction of the optimal $t_{\max}$ value for an arbitrary RMC calculation is an unsolved problem. In any case $t_{\max}$ can be obtained empirically by trial and error method. Note that the increase of the maximal allowed displacements of the atoms from their initial positions or the increase of the number of variables (i.e., the number of atoms) usually lead to an **exponential** growth of the required computational time.

The simulated annealing scheme, implemented in EvAX, resembles Boltzmann annealing schedule. It ensures linear decrease of the acceptance probability p_0 during the simulation: at the beginning of the simulation $p_0(0)$ is set to the value speciefed by the **Acceptance_rate** parameter, while at the end of simulation $p_0(t_{\max})$=0. In between p_0 changes linearly: $p_0(t) = p_0(0)(1-t/t_{\max})$.

Thus in this scheme only one parameter - length of cooling schedule $t_{\max}$ - needs to be specified. It can be done using parameter **Froze_in**. For example,

```
adjust_acceptance    1
Froze_in             1500
Acceptance_rate      0.8
```

sets $t_{\max}$ to 1500 iterations and $p_0(0)$ to 0.8. For simple systems this value can be sufficient. For more complex systems, where the optimal solution may be quite far from the initial structure, $t_{\max}$ values up to 100000 and higher may be required.

Note that calculations are not stopped after $t_{\max}$ iterations. This value just sets the moment, after which only atomic displacements that are improving agreement between experiment and theory are accepted. As long as such moves can be found, calculations will be continued (see Sec. 6.11 for the discussion on stopping criterion).

6.10 Evolutionary algorithm

One of the main strengths of EvAX code is that evolutionary algorithm (EA) is employed for structure model optimization, which makes the search for the optimal structure model much more efficient that it is in the conventional RMC method.

Evolutionary algorithm is a general name for a class of similar, population-based techniques for parameter optimization that strive to mimic genetic processes in the natural systems. The most well-known type of EA is genetic algorithm, introduced by John Holland in the 1960s [36, 37, 38, 39], for which the encoding of the variables in the form of binary strings is characteristic.

In the EA a set of current values of all parameters, which should be optimized, forms an "individual". The function of these parameters, which needs to be maximized, is called fitness function. The "population" is an ensemble of such "individuals": it consists of many sets of parameters having different values. At each iteration some manipulations with the population are carried out, so that a new population is obtained from the old one. If these manipulations, or operators, applied to the old population, are properly chosen, the fitness of the new population will be, on average, larger than that for the old population. Thus, the fitness function will be maximized step by step, and the values of parameters will approach those at the global maximum point.

When EA is applied to the interpretation of EXAFS spectra, the variable parameters are the coordinates of the atoms. An individual hence is an atomic configuration. For such individual one can perform ab initio EXAFS calculations, and compare the obtained theoretical result with the experimental one. The fitness function is defined so that it has the maximal value, when the difference between the theoretical and experimental EXAFS spectra is the smallest. Thus, the maximization of the fitness function is equivalent to the search for the atomic configuration, which represents the atomic structure of the sample and for which the theoretical EXAFS spectrum is close to the experimental one. Using the difference between the theoretical and experimental EXAFS data ξ, already introduced for the RMC algorithm, the fitness function, required for the EA, can be simply defined as $-\xi$.

In the EA three operators can be applied to a given population: (i) selection, (ii) crossover and (iii) mutation (Fig. 6.10). In the selection process one creates a new population (i.e., a new set of atomic configurations) from the individuals of the old population, where the individual with larger fitness function has larger probability to get into the new population. Thus, there will be less atomic configurations with low values of fitness function in the new set, but instead there can be several copies of the atomic configurations with high values of the fitness function. In the crossover process one replaces two randomly selected atomic configurations ("parents") from the old population with two other atomic configurations ("children"), obtained by interchanging some parts of the "parents", i.e., we replace some atoms of one atomic configuration with atoms of the other configuration. Finally, the mutation operator applies some small and random changes to all individuals of the population: in our case, all atoms in all configurations are slightly displaced in random directions.

These three basic operators are implemented in the EvAX code, and will be discussed below.

The convergence properties of EA is a subject of significant controversy. It is clear, however, that at least for some optimization problems EA is much more efficient tool than the conventional simulated annealing. Intuitively it can be understood using following argument: let us assume that the fitness function $-\xi(x_1,x_2,...,x_k)$ that is a function of variables x_1, x_2, ..., x_k can be decomposed as $-\xi(x_1,x_2,...,x_k) = -\xi_1(x_{11},x_{12},...) - \xi_2(x_{21},x_{22},...) - ... - \xi_j(x_{j1},x_{j2},...)$. Here $(x_{11},x_{12},...)$, $(x_{21},x_{22},...)$, etc., are some (not necessarily non-overlapping) subsets of the original set of variables $(x_1,x_2,...,x_k)$. In the case of EA approach we have a population of several replicas of our system. At each iteration we select the fittest (i.e., the ones having the largest value of $-\xi$) individuals to the next generation. Some of the selected individuals will be included in this next generation because they have large value of, say, $-\xi_1$, some others - due to the high value of $-\xi_2$, etc. Thus we may formulate that instead of the maximization of function $-\xi$ of k variables, EA is solving several maximization problems, but each with a fewer number of variables [39]. As was pointed out previously, the number of iterations, required to find the optimal solution to the problem using, for instance, simulated annealing, is exponentially dependent on the number of variables k. So if we are able to factorize the fitness function so that all sets $(x_{j1},x_{j2},...)$ have a significantly fewer number of variables than k, the number of required iteration to find the optimal solution will decrease exponentially with respect to the conventional simulated annealing: in this case EA will be much more efficient.

It appears, the case of EXAFS analysis is one of those, where EA methods can be useful: the total EXAFS spectrum can be expressed as a sum of contributions from relatively small atomic clusters, and these clusters act like the "genes" of population. The "genes" that ensure high values of the fitness function survive from one generation to another, and an exchange of the "genes" between individuals efficiently maximizes the fitness of each individual.

As shown in [4], in some cases EA approach allows almost exponential reduction of the number of EXAFS recalculations, hence - almost exponential reduction of required computational time, with respect to conventional RMC-EXAFS method.

To enable optimization via evolutionary algorithm, use **Number_of_states** parameter, which specifies number of individuals in the population. By setting **Number_of_states** to a value larger than 1, optimization will be carried out via EA scheme. For example,

```
Number_of_states    32
```

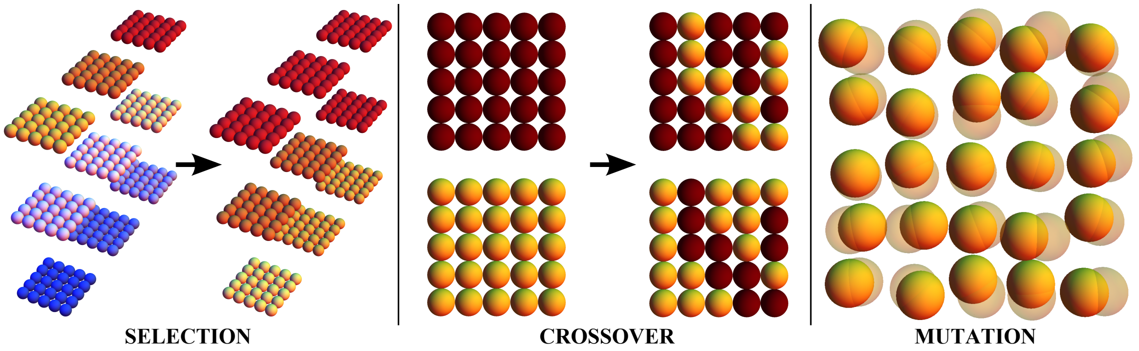

Figure 6.1: Three basic operators of evolutionary algorithm
Selection operator chooses the best of individuals to the next generation, keeping the total number of individuals in the population constant; crossover operator interchanges some parts of individuals in the population, while mutation operator applies some random changes to individuals.

starts EA simulations with 32 individuals in population.

```
Number_of_states     1
```

in turn, starts conventional RMC simulations.

Good results and computational efficiency typically are achieved with **Number_of_states** values between 16 and 256.

Selection

The selection operator can be implemented using different procedures: for example, the conventional proportional selection (the probability of the individual to be selected to the new generation is proportional to the value of its fitness function), truncation selection (the individuals are ranged accordingly to their fitness, and fixed number of them is selected to the new generation), etc. [40, 41]

In EvAX we use the so-called tournament selection: two individuals from the old population are randomly selected, and the one with higher fitness is copied to the new population, and the process is repeated until the number of individuals in the new populations is the same as in the old population [40]. The advantages of this scheme are that (i) it is easy to implement, and (ii) it better preserves the "genetic diversity" of the population than proportional selection. In the case of proportional selection the probability of individual with low fitness to be selected to the new population can be extremely small (if, for example, the fitness of the best individual is much higher than the fitness of all other individuals), and, therefore, all individuals will be close to each other after relatively small number of iterations, thus all advantages, provided by the use of selection and crossover operators, will be lost.

Crossover

For crossover operator we split in pairs all individuals in the population, which is obtained after applying the selection operator, and then perform the "breeding" of each pair with a given probability (currently a fixed value of 50% is used). First, a pseudo-random number ρ_b ($0 \leq \rho_b \leq 1$) is generated. If $\rho_b > 0.5$, the old pair of individuals ("parents") is replaced with a pair of new individuals ("children"). To create "children", we generate a set of pseudo-random numbers ρ_a ($0 \leq \rho_a \leq 1$) for each atom in the supercell. If $\rho_a < 0.5$, the corresponding atom of the first of the children is set at the same position as the corresponding atom of the first of the parents, and the corresponding atom of the second of the children is set at the same position as the corresponding atom of the second of the parents. If $\rho_a > 0.5$, the interchange of atoms takes place: corresponding atom of the first of the children now is taken from the second of the parents, and the corresponding atom of the second of the children is taken from the first of the parents.

As the crossover changes the atomic configuration, the corresponding EXAFS spectrum should be recalculated for the bred individuals, thus increasing the number of total EXAFS recalculations required. However, an advantage of the use of the crossover operator for EXAFS analysis is based on the fact that EXAFS spectra can be expressed as a series of contributions from separate photoelectron paths, where each path contains only a few atoms. After crossover operation some of these paths are modified, but a large number of them remains unchanged, therefore one can expect, the "children" of the "parents" with good fitness will also have high values of fitness function.

Mutation

In the conventional EA the mutation is a fully random change of the individuals, i.e., one should apply random displacements to all atoms of the population. In our approach, however, we apply additional restriction on the mutations, defined by the Metropolis criterion (Eq. (6.1)) and simulated annealing algorithm. One mutation step in EA approach thus is completely equivalent to a conventional RMC step.

Diversity

The optimization efficiency improvement, provided by evolutionary algorithm, depends on how diverse is the set of atomic configurations, used for selection and crossover. Clearly, if all atomic configurations are the same, selection and crossover will not be able to improve the agreement between experiment and theory. Therefore diversity is an important quantity to keep an eye on.

In our case we can define the diversity of atomic configurations as

$$\sigma_d^2 = \sum_i \langle \Delta |\vec{v}_i|^2 \rangle - |\langle \Delta \vec{v}_i \rangle|^2, \tag{6.3}$$

where $\Delta\vec{v}_i$ is the deviation of the i-th atom from its position in the initial structure model, and we average such deviations over all atomic configurations, used for evolutionary algorithm.

Note that σ_d^2 will gradually decrease during EvAX simulations, since selection and crossover operators will try to make the atomic configurations more and more similar, and the efficiency of mutation operator will be reduced by reduction of the number of approved atomic displacements ensured by simulated annealing.

6.11 End of EvAX simulations

As mentioned before, after number of iteration reach the value, specified by **Froze_in** parameter, only the atomic displacements that are improving agreement between experiment and theory are accepted. Clearly, the number of such displacements is limited, therefore soon after **Froze_in** iterations are done, the structure model will not evolve via mutation procedure. Selection and crossover operations then quickly will make all structure models (individuals) very close to each other, and, after this happens, structure model will not improve anymore.

To detect, if the structure model is improving or not, EvAX compares the difference between experimental and simulated data ξ after each **time2** iterations. If number of iterations is larger than **Froze_in** and after **time2** the ξ value has not improved, EvAX calculations will stop.

Alternatively, one can force EvAX to stop calculations using **Stop_after** parameter. For example,

```
Stop_after   10000
```

will force EvAX to stop after 10000 iterations, independently on **Froze_in** value. To rely instead on the stopping criterion, discussed above, set **Stop_after** parameter to any negative value. For example,

```
Stop_after   -1
Froze_in     1500
time2        100
```

will force EvAX to stop calculations, if number of iterations has reached 1500, and within 100 iterations after that moment no improvement in ξ value was detected.

7. Output and results

7.1 Files with structure

At the first iteration, after the file with the initial structure description (e.g., with the fractional coordinates of equilibrium structure) is imported, EvAX saves the complete list of Cartesian atomic coordinates in a file with the name, specified as **File_for_equilibrium_xyz**. Standard XYZ chemical file format is used here: first line contains total number of atoms, second line is comment line, and the remaining rows in the file each corresponds to one atom, and contains its label, and x, y and z coordinates.

Importantly, if periodic boundary condition are used (**pbc** = 1), the number of atoms in this file will be 27 times larger than the actual number of atoms in the structure model N_{at}. While the first N_{at} in **File_for_equilibrium_xyz** file will correspond to "real" atoms, the remaining $26 \times N_{\mathrm{at}}$ rows of the file will contain coordinates of copies of the first N_{at} atoms, translated by vectors $\vec{v}_{ijk} = in_a\vec{a} + jn_b\vec{b} + kn_c\vec{c}$, where $\vec{a}$, $\vec{b}$ and $\vec{c}$ are unit cell vectors specified in the file with the structure, n_a, n_b and n_c are the integers that define the supercell size (specified in the parameter file as **Supercell** parameter), and i, j k each assumes values 0, -1, 1. These copies of atoms are useful to avoid surface effects in the analysis of the structure model, for example, when atomic radial distribution function is calculated.

In addition to a file with the name specified by **File_for_equilibrium_xyz** parameter, EvAX will create also an XYZ file with the same name but with *..._no_PBC* added. This second file contain the same atomic coordinates, but does not contain the copies of atoms created by PBC. This file is useful to visualize the atomistic structure using programs like *Vesta*.

The same format is used also to store the current atomic coordinates during EvAX simulations. The name of the file with current atomic coordinates is specified as **File_for_final_xyz** parameter. The order of atoms in the **File_for_final_xyz** file is exactly the same as the order of atoms in **File_for_equilibrium_xyz** file, which allows one to identify easily all atoms in the final structure model.

Parameter **time2** determines, how often the file with the current atomic coordinates is updated (coordinates are saved to disk only after each **time2** iterations). For example,

time2	**100**
File_for_equilibrium_xyz	**start.xyz**
File_for_final_xyz	**final.xyz**

specifies that initial atomic configuration will be saved to "*start.xyz*" file, while the current positions of atoms will be saved to "*final.xyz*" file, which will be updated every 100 iterations.

Note that if evolutionary algorithm approach is used and several atomic configurations are modeled simultaneously, only the atomic configuration that gives the best agreement with experimental data is saved. To follow, how the atomic configuration evolves during the simulations, and to get coordinates for other atomic configurations, one can use the information from the restart files (see Section 7.4).

7.2 Files with spectra

Before the main EvAX run, experimental EXAFS spectra are processed. First, they are re-interpolated on the k-grid with k-values between **k_min** and **k_max**, specified in parameter file, and k-step **dk**. Experimental spectrum is shifted by **dE0**, and divided by **S02**, as explained in Chapter 3. The re-interpolated and corrected experimental spectrum is then saved to disk with the file name, specified as **File_for_i-pol._exp._signal** parameter in the parameter file. The order of columns with k values and EXAFS signal $\chi(k)k^n$ values in this output file is the same as the order of these columns in the file with original experimental EXAFS data, and also the used power n in k^n factors is the same as in the original file (as determined by **Column_for_k** , **Column_for_chi** and **k_power_in_IO** parameters).

Next, Fourier transform, inverse Fourier transform and Morlet wavelet transform for experimental data are carried out, and results of these transformations are saved to disk. The names of the files are specified as **File_for_FT_of_exp._signal**, **File_for_BFT_of_exp._signal** (for Fourier transform and inverse Fourier transform) and **File_for_WT_of_exp._signal** (for wavelet transform).

Files with Fourier transform data have 4 columns: the first contains the values of the interatomic distances (frequencies) R, the 2nd, 3rd and 4th contain correspondingly real part, imaginary part and modulus of Fourier transform.

Files with wavelet transform data have 5 columns: the first contains the values the wavenumber k, the second contains the values the interatomic distances (frequencies) R, while the 3rd, 4th and 5th columns contain corresponding real part, imaginary part and absolute value of wavelet transform.

Files with inverse Fourier transform has 2 columns: the first contains the values the wavenumber k, the second contains the values the of the Fourier-filtered EXAFS spectrum.

For example,

File_for_i-pol._exp._signal	**ipolEXAFS.dat**
File_for_FT_of_exp._signal	**expFT.dat**
File_for_BFT_of_exp._signal	**expBFT.dat**
File_for_WT_of_exp._signal	**expWT.dat**

specifies that files "*ipolEXAFS.dat*", "*expFT.dat*", "*expBFT.dat*" and "*expWT.dat*" should be used to save re-interpolated experimental EXAFS signal, its Fourier, inverse Fourier and wavelet transform.

Similarly stored are also the theoretical EXAFS spectra. Files with the calculated spectrum, its Fourier transform, inverse Fourier transform and wavelet transform are specified as parameters

File_for_calculated_EXAFS_signal, **File_for_FT**, **File_for_BFT** and **File_for_WT**, correspondingly, for example

```
time1                             10
File_for_calculated_EXAFS_signal  EXAFS.dat
File_for_FT                       FT.dat
File_for_BFT                      BFT.dat
File_for_WT                       WT.dat
```

Parameter **time1** determines, how often the files with theoretically calculated data are updated: note that while EXAFS spectra are recalculated at each iteration, they are saved to disk only after each **time1** iterations. If evolutionary algorithm approach is used and several atomic configurations are modeled simultaneously, only the spectra that correspond to the configuration that gives the best agreement with experimental data are saved. To follow, how the spectra evolve during the simulations, and to get access to the spectra for other atomic configurations, one can use the information from the restart files (see Section 7.4).

In addition, it is often useful to have access not only to the total calculated EXAFS spectrum, but also to the partial contributions from each of the photoelectron paths. These partial spectra are saved to a file with the name specified as **File_for_partial_EXAFS**, e.g.,

```
time2                   100
File_for_partial_EXAFS  partialEXAFS.dat
```

The first column of this file contains k-values, while each of the remaining columns - corresponding $\chi_p(k)k^n$ value, where $\chi_p(k)$ is EXAFS spectrum, calculated for the p-th path. The order of paths is the same as in "*paths_fin.dat*" file in "feff_0" directory. Note that **File_for_partial_EXAFS** file is updated every **time2** iterations (typically **time2** > **time1**).

7.3 Fit status

The most relevant information on the current fit status is printed to the standard output every **time1** iterations, as well as saved to a file, whose name is specified in parameters file as **File_for_residuals**. E.g.,

```
time1               10
File_for_residuals  status.dat
```

will make EvAX to update every 10 iterations the file "*status.dat*" with the information on the current difference between experimental and theoretical EXAFS data. This file can be used, e.g., to monitor the convergence of the EvAX calculations.

The format of this file can be deduced from its header (first line of the file). The first column contains the number of corresponding iteration, while the 2nd column (`[Best_Res._(xi)]`) - the difference between experimental and theoretical spectra for configuration that at that iteration gave the best agreement with experiment. Third column is currently not used and contains 0. The 4th column (`Best_Objective`) contains the same information as the 2nd column, but is normalized by the average change of difference between experiment and theory (see Eq. (6.2)). The average change of difference between experiment and theory itself is reported in the 5th column `[Change_(Delta)]`. 6th column (`[Acceptance]`) contains the logarithm of the target acceptance rate p_0 (see Eq. (6.2)). 7th column (`[EA_diversity]`) contains information about the diversity σ_d^2 of atomic configurations, used for evolutionary algorithm simulations (see Eq. (6.3)). Remaining columns of this file contain information about the difference between experiment and theory for each of the atomic configurations, used for evolutionary algorithm simulations.

Similar information is printed out also to the standard output. Every **time1** iterations, EvAX prints out current iteration number (`Generation`) and the difference between experimental and theoretical spectra for configuration that at that iteration gave the best agreement with experiment (`BEST-STATE RESIDUAL`). Note that differences in $k-$, $q-$, $R-$ and wavelet ($w-$) spaces all are reported. In addition, if several spectra (e.g., EXAFS data acquired at two absorption edges) are fitted simultaneously, partial differences between experimental and theoretical data will also be printed out. Difference between experiment and theory, normalized by the average change of difference between experiment and theory, is reported next as `BEST-STATE OBJECTIVE`, as well as the evolutionary algorithm diversity σ_d^2 (`EA DIVERSITY`). Finally, the average change of difference between experiment and theory is reported as `REGULATION / Delta`, logarithm of target acceptance rate is reported as `REGULATION / Log[acceptance]`, while the actual ratio of discarded atomic displacements during the last **time1** iterations is reported as `REGULATION / Discarded` values.

7.4 Restart files

Restart files are created with the frequency, set by parameter **time2**. The name of restart file is a string, set by parameter **File_for_restart**, to which an integer number, corresponding to the number of current iteration, is appended.

Restart files are text-type files that consist of two large blocks: the first block has the same structure as the input parameter file, where all the current values of system parameters are given. Thus any restart file, produced by EvAX, can be used directly to start a new calculation.

The second block starts after keyword `END` and contains information on the EvAX version, used to generate the restart file, number of current iteration, and then lists current value of differences between experimental and calculated EXAFS spectra. Note that if evolutionary algorithm approach is used and several atomic configurations are modeled simultaneously, values of differences between experiment and theory for all configurations are reported. Next line (`Average_delta`) contains the average change of difference between experiment and theory per one iteration, used by simulated annealing algorithm. Next line (`Average_delta_E`) currently is not used.

Next follows keyword `ATOMS`, after which coordinates of all atoms in all atomic configurations in XYZ format are listed.

Finally, after keyword `SPECTRA`, current calculated EXAFS spectra are recorded. Again, if evolutionary algorithm approach is used and several atomic configurations are modeled simultaneously, EXAFS spectra for all configurations are reported. Only $\chi(k)k^n$ values are given. Corresponding k-values are not reported, but they span the k-range from **k_min** to **k_max** with the step **dk**, as specified in the parameters file.

Thus EvAX calculations can be restarted from the point, at which the corresponding restart file was created. For this purpose run command like

```
EvAX.exe   restart1000 -restart
```

Here "*restart1000*" is the name of the restart file. Note, however, that EXAFS spectrum, calculated immediately after restart of simulations, can be slightly different from the spectrum, as saved to restart file due to the approximations, discussed in Chapter 4. Usually this discrepancy does not affect the results.

If evolutionary algorithm approach is used, it is sometimes necessary to extract the atomic coordinates and spectra, corresponding only to one atomic configuration that gives the best agreement between experiment and theory. For this purpose, one can use **-beststate** command. For example,

```
EvAX.exe    restart1000 -beststate
```

will create a copy of "restart1000" file with the name "restart1000_beststate_1000", where only atomic coordinates and EXAFS spectrum for the best configuration will be reported. This file is a valid restart file and can be used to run EvAX calculations in restart mode.

7.5 Quick analysis of final structure

The main result yielded by EvAX is a set of atomic coordinates, corresponding to the final atomic configuration. It should be emphasized that such solution is, of course, not unique. If one repeats the simulations with the same input parameters, but employing different sequences of pseudo-random numbers, the final set of atomic coordinates will be different. However, the statistical characteristics, such mean values and dispersions of interatomic distances and bond angles, distribution functions for distances and angles, should be close for both cases, and also close to their "experimental" analogues, if the agreement between calculated EXAFS spectra and experimental data is sufficiently good. Hence the post-processing of the results is important to extract physically relevant information.

The sets of interatomic distances, bonding angles and more complex structural elements, corresponding to the final configuration of supercell, can be considered as statistical samples that can be used to estimate the distributions of such elements in the population, i.e., in the investigated material. Probably the most important in our case are the distributions of interatomic distances, or so called radial distribution functions (RDFs), since they influence the EXAFS spectra through the single-scattering contributions that usually are dominant.

The easiest way to estimate the RDFs or distribution functions of bonding angles, etc., is to construct a histogram of the corresponding quantity in the statistical sample. In this case no *a-priory* knowledge on the shape of the corresponding distribution function is necessary. EvAX program contains a rudimentary built-in tool that calculates bond lengths histograms for the final structure. RDFs are saved to the file, specified as **File_for_partial_RDFs**, while the histogram bin width (resolution) is set by **RDF_bin_width** parameter. For example,

```
RDF_bin_width           0.01
File_for_partial_RDFs   rdf.dat
```

saves to disc "*rdf.dat*" text file that contains the partial RDFs for all types of atoms, parametrized as

histograms with bin width 0.01 Å. Partial RDFs for different types of atoms are saved to different columns of the created text file, and can be identified by the header line. The obtained RDFs can then be plotted by any graphical software.

To obtain more quantitative information (average interatomic distances, mean-square relative displacement (MSRD) factors, etc.), one needs to estimate mean values, variances, in some cases -

even higher moments of the investigated distributions. While the estimation of the mean value of population, having a representative statistical sample, is not a particular problem - the consistent and unbiased estimator of the population mean is just the mean value of the sample - the situation is more complicated with higher moments. The structure models that we are using for our calculations are relatively small. Therefore the variance (and, of course, higher moments) of the statistical sample are extremely sensitive to the presence of any outliers in our data and cannot be used as estimators of the corresponding quantities of population.

To solve this problem one needs to use more robust techniques. If the shape of the distribution is known (for instance, if it can be described by a Gaussian function), one can simply fit the obtained histogram of the statistical sample with the corresponding density distribution function, and hence simultaneously estimate all parameters of interest. Obviously, this approach will give a consistent estimation of structure parameters only if the underlying distribution indeed is, say, Gaussian. Sometimes this property can be even advantageous: for example, the conventional EXAFS analysis often is carried out in the Gaussian approximation. Therefore, using the fitting approach to process the RDFs from EvAX simulations, the obtained results are directly comparable with the results of conventional EXAFS analysis.

Alternative approach to estimate variance is so called **median absolute deviation** (MAD) method [42, 43]. The standard deviation σ of a distribution in this case is estimated as

$$\sigma_{\mathrm{MAD}} = \mathrm{median}\left(|x_1 - m|, |x_2 - m|, ..., |x_n - m|\right), \tag{7.1}$$

where $(x_1, x_2, ..., x_n)$ is the statistical sample and $m = \mathrm{median}(x_1, x_2, ..., x_n)$. The advantage of the MAD approach is that it can be applied to characterize the deviations of data, even if the shape of analyzed distribution is unknown. At the same time, MAD is an inconsistent estimator, i.e., even if the analyzed sample is infinitely large, the MAD estimator will not be equal to the true value of the standard deviation of population. On the other hand, if the approximate shape of the investigated distribution is known, one can convert MAD in a consistent estimator of σ, by simply multiplying the MAD value σ_{MAD} by some scaling constant k that, noteworthy, is independent on the sample size. For Gaussian distribution $k = 1.4826$ [43]. For other distributions the corresponding value of k can be derived analytically or obtained in numerical simulations.

Related to the problem of determination of MSRD factors is also the determination of mean-square displacement (MSD) factors: in this case we are interested in the mean-square displacement of atom from its equilibrium position $\langle u^2 \rangle$. By analogy with the previous case, we can estimate this quantity as

$$\langle u^2 \rangle = (k \times \mathrm{median}(u))^2. \tag{7.2}$$

In the case of isotropic and harmonic atomic oscillations, atomic displacements u are distributed according to Maxwell distribution $f(u) = u^2 e^{-u^2/(2\sigma^2)}/(\sqrt{\pi/2}\sigma^3)$. The sought quantity $\langle u^2 \rangle$ then is equal to $\int_0^\infty f(u) u^2 du = 3\sigma^2$. Median of the distribution, in turn, is equal to $\mathrm{median}(u) = \sigma\sqrt{2Q^{-1}(3/2, 1/2)} \approx 1.538\sigma$, where $Q(a,b) = \Gamma(a,z)/\Gamma(a)$ is regularized incomplete gamma function. Thus the the scaling coefficient k in Eq. 7.2 is equal to $k = \sqrt{3}/1.538 = 1.126$.

Average interatomic distances and average interatomic angles, MAD estimators of bond length variances and angle variances and approximate values for isotropic MSD factors are reported by EvAX to the standard output at the end of simulations.

First, MSD factors for each of the different atom types are reported (calculated via Eq. 7.2). It is important to emphasize that to calculate MSD, instead of actual atomic displacements from *equilibrium positions*, EvAX will use the displacements from *initial positions*. If the initial positions were close to the equilibrium positions (as determined, e.g., by diffraction methods), the obtained MSD value will be accurate. In other cases one may need to carry out more advanced analysis of atomic coordinates, generated by EvAX, to calculate MSD reliably. Note also that for many materials the atomic vibrations are significantly anisotropic, and in this case the isotropic MSD factors, provided by EvAX, may be not too informative, and, again, some additional analysis of atomic coordinates may be necessary.

Next, the average interatomic distances, MSRD factors, average bonding angles and their variances are reported. For this purpose EvAX uses format similar to that in "*paths.dat*", generated by FEFF. The list of most important single scattering and double-scattering paths is printed out. For each paths then the average interatomic distance (`<rleg>`), average bonding angle (`<beta>`), MAD MSRD factor (`MAD[rleg]`, calculated as $(1.4826\sigma_{\mathrm{MAD}})^2$, and MAD estimator of bonding angle variance (`MAD[beta]`, also calculated as $(1.4826\sigma_{\mathrm{MAD}})^2$, is reported.

Example of such output for crystalline Ge is shown below. It contains information about MSD factor for Ge atoms (0.00617047 Å^2), average interatomic distances within the first three coordination shells (2.452395, 4.002310 and 4.692654 Å, respectively) and corresponding MSRD factors (0.002304, 0.003657 and 0.004392 Å^2, respectively), as well as average Ge–Ge–Ge bonding angles (70.578198 and 144.710901 degrees) and their variances (6.279562 and 1.975570 deg^2).

```
------------------------------------------------------------
-----------------------------------------------------------
09.03.2015 11:57:22              MSD <u^2>
-----------------------------------------------------------
Ge 0.00617047
-----------------------------------------------------------
09.03.2015 11:57:22              STRUCTURE PARAMETERS
-----------------------------------------------------------
     1    2   4  index, nleg, degeneracy, r=  2.449986
 x        y        z        ipot label <rleg>    <beta>    MAD[rleg] MAD[beta]
 1.414500 1.414500 1.414500    1    Ge 2.452395 180.000000 0.002304 0.000000
 0.000000 0.000000 0.000000    0    Ge 2.452395 180.000000 0.002304 0.000000
     2    2   12  index, nleg, degeneracy, r=  4.000810
 x        y        z        ipot label <rleg>    <beta>    MAD[rleg] MAD[beta]
 0.000000 2.829000 2.829000    1    Ge 4.002310 180.000000 0.003657 0.000000
 0.000000 0.000000 0.000000    0    Ge 4.002310 180.000000 0.003657 0.000000
     3    3   12  index, nleg, degeneracy, r=  4.450391
 x        y        z        ipot label <rleg>    <beta>    MAD[rleg] MAD[beta]
 1.414500 -1.414500 -1.414500    1    Ge 2.452395 144.710901 0.002304 1.975570
 1.414500 1.414500 1.414500    1    Ge 4.002310 144.710901 0.003657 1.975570
 0.000000 0.000000 0.000000    0    Ge 2.452395 70.578198 0.002304 6.279562
     4    3   24  index, nleg, degeneracy, r=  4.450391
 x        y        z        ipot label <rleg>    <beta>    MAD[rleg] MAD[beta]
 0.000000 2.829000 2.829000    1    Ge 4.002310 144.710901 0.003657 1.975570
 1.414500 1.414500 1.414500    1    Ge 2.452395 70.578198 0.002304 6.279562
```

```
0.000000 0.000000 0.000000    0    Ge 2.452395 144.710901 0.002304 1.975570
    5    2   12  index, nleg, degeneracy, r=  4.691366
x        y        z        ipot label <rleg>    <beta>    MAD[rleg] MAD[beta]
4.243500 1.414500 -1.414500    1    Ge 4.692654 180.000000 0.004392 0.000000
0.000000 0.000000 0.000000    0    Ge 4.692654 180.000000 0.004392 0.000000
```

Note finally that even in the ideal case of model data without any experimental noise due to the limited size of the sample the parameters of the distribution will always be estimated within some statistical uncertainty. In the case of EvAX simulations such statistical uncertainty can be assessed by comparing the results obtained in simulations with different sequences of pseudo-random numbers and/or different initial conditions.

Bibliography

[1] J. J. Rehr and R. C. Albers. Theoretical approaches to x-ray absorption fine structure. *Rev. Mod. Phys.*, 72:621, 2000.

[2] S J Gurman and R L McGreevy. Reverse Monte Carlo simulation for the analysis of EXAFS data. *J. Phys.: Condens. Matter*, 2(48):9463, 1990.

[3] J. Timoshenko, A. Kuzmin, and J. Purans. Reverse monte carlo modeling of thermal disorder in crystalline materials from exafs spectra. *Comput. Phys. Commun.*, 183(6):1237, 2012.

[4] J Timoshenko, A Kuzmin, and J Purans. Exafs study of hydrogen intercalation into reo 3 using the evolutionary algorithm. *Journal of Physics: Condensed Matter*, 26(5):055401, 2014.

[5] A. L. Ankudinov, B. Ravel, J. J. Rehr, and S. D. Conradson. Real-space multiple-scattering calculation and interpretation of x-ray-absorption near-edge structure. *Phys. Rev. B*, 58:7565, September 1998.

[6] Koichi Momma and Fujio Izumi. Vesta 3 for three-dimensional visualization of crystal, volumetric and morphology data. *J. Appl. Crystallogr.*, 44(6):1272, 2011.

[7] B Ravel and M Newville. Athena, artemis, hephaestus: data analysis for x-ray absorption spectroscopy using ifeffit. *J. Synchrotron Radiat.*, 12(4):537, 2005.

[8] A. Kuzmin. EDA: EXAFS data analysis software package. *Physica B*, 208-209:175, 1995.

[9] A Filipponi, A Di Cicco, and C R Natoli. X-ray-absorption spectroscopy and n-body distribution functions in condensed matter. i. theory. *Phys. Rev. B*, 52(21):15122, 1995.

[10] R. L. McGreevy and L. Pusztai. Reverse Monte Carlo simulation: a new technique for the determination of disordered structures. *Mol. Simul.*, 1(6):359, 1988.

[11] Joshua Kas. *Toward Quantitative Calculation and Analysis of X-Ray Absorption Near Edge Spectra*. PhD thesis, University of Washington, 2009.

[12] J. J Rehr, A. L. Ankudinov, and B. Ravel. FEFF8.40 User Guide, 2006.

[13] Mahdi Ahmadi, Janis Timoshenko, Farzad Behafarid, and Beatriz Roldan Cuenya. Tuning the structure of pt nanoparticles through support interactions: an in situ polarized x-ray absorption study coupled with atomistic simulations. *The Journal of Physical Chemistry C*, 123(16):10666–10676, 2019.

[14] Scott Calvin. *XAFS for Everyone*. CRC Press, 2013.

[15] IXS Standards and Criteria Committee. Error reporting recommendations: A report of the standards and criteria committee, 2000.

[16] S. Muto. Application of spline wavelet transformation to the analysis of extended energy-loss fine structure. *J. Electron Microsc.*, 49(4):525, 2000.

[17] Limin Shao, Xiangqin Lin, and Xueguang Shao. a wavelet transform and its application to spectroscopic analysis. *Appl. Spectrosc. Rev.*, 37:429, 2002.

[18] M. Muñoz, P. Argoul, and F. Farges. Continuous cauchy wavelet transform analysis of exafs spectra: A qualitative approach. *Am. Mineral.*, 88(4):694, 2003.

[19] H. Funke, AC Scheinost, and M. Chukalina. Wavelet analysis of extended x-ray absorption fine structure data. *Phys. Rev. B*, 71(9):094110, 2005.

[20] G.E. Brown, J.G. Catalano, A.S. Templeton, T.P. Trainor, F. Farges, B.C. Bostick, T. Kendelewicz, C.S. Doyle, A.M. Spormann, K. Revill, et al. Environmental interfaces, heavy metals, microbes, and plants: applications of xafs spectroscopy and related synchrotron radiation methods to environmental science. *Phys. Scripta*, 2005:80, 2005.

[21] M. Munoz, F. Farges, and P. Argoul. Continuous cauchy wavelet transform of xafs spectra. *Phys. Scripta*, 2005:221, 2005.

[22] M. Harfouche, E. Wieland, R. Dähn, T. Fujita, J. Tits, D. Kunz, and M. Tsukamoto. Exafs study of u (vi) uptake by calcium silicate hydrates. *J. Colloid Interf. Sci.*, 303:195, 2006.

[23] M. Vespa, R. Dähn, D. Grolimund, E. Wieland, and AM Scheidegger. Spectroscopic investigation of ni speciation in hardened cement paste. *Eviron. Sci. Technol.*, 40(7):2275, 2006.

[24] H. Funke, M. Chukalina, and A.C. Scheinost. A new feff-based wavelet for exafs data analysis. *J. Synchrotron. Radiat.*, 14(5):426, 2007.

[25] H. Funke, M. Chukalina, and AC Scheinost. Application of wavelet transform to extended x-ray absorption spectroscopy. In *Proceedings of the 7th WSEAS international conference on Wavelet analysis & multirate systems*, page 117. World Scientific and Engineering Academy and Society (WSEAS), 2007.

[26] M. Sahnoun, C. Daul, and O. Haas. Wavelet investigation of $la_{0.5}ca_{0.5}coo_{3-\delta}$ x-ray absorption data. *J. Appl. Phys.*, 101(1):014911, 2007.

[27] Charles K Chui. *An introduction to wavelets*, volume 1. Academic press, 1992.

[28] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[29] R. L. McGreevy and P. Zetterström. Reverse Monte Carlo modelling of network glasses: useful or useless? *J. Non-Cryst. Solids*, 293-295:297, 2001.

[30] Makoto Matsumoto and Takuji Nishimura. Mersenne Twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3, January 1998.

[31] Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, J?rgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. Copasi?a complex pathway simulator. *Bioinformatics*, 22(24):3067, 2006.

[32] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J Chem. Phys.*, 21:1087, 1953.

[33] Onuttom Narayan and AP Young. Convergence of monte carlo simulations to equilibrium. *Phys. Rev. E*, 64(2):21104, 2001.

[34] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671, May 1983.

[35] L. Ingber and B. Rosen. Genetic algorithms and very fast simulated reannealing: A comparison. *Math. Comput. Model.*, 16(11):87, 1992.

[36] D.E. Goldberg and J.H. Holland. Genetic algorithms and machine learning. *Mach. Learn.*, 3(2):95, 1988.

[37] T. Bäck and H.P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.*, 1(1):1, 1993.

[38] J.H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press, 1992.

[39] Darrell Whitley. An overview of evolutionary algorithms: practical issues and common pitfalls. *Inform. Software Tech.*, 43(14):817, 2001.

[40] D. Thierens and D. Goldberg. Convergence models of genetic algorithm selection schemes. *Lect. Notes Comput. Sc.*, 866:119, 1994.

[41] T. Blickle and L. Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evol. Comput.*, 4(4):361, 1996.

[42] F.R. Hampel. Robust estimation: A condensed partial survey. *Probab. Theory Relat. Field*, 27(2):87, 1973.

[43] M. Daszykowski, K. Kaczmarek, Y. Vander Heyden, and B. Walczak. Robust statistics in data analysisa review: basic concepts. *Chemometrics Intell. Lab. Syst.*, 85(2):203, 2007.