AUTOBK

Automated Background Removal for XAFS Data

AUTOBK was written with the guidance and encouragement of Edward A. Stern at the University of Washington. The idea for the main algorithm used in AUTOBK (that is, adjusting the background until the low-frequency components of the resulting XAFS match those of a model calculation) was first suggested to me by Yizhak Yacoby. With great help from Pēteris Līviņš, Ed Stern, and Yizhak Yacoby, AUTOBK evolved from this algorithm, the application of information theory to XAFS analysis, and the desire for an automated way to reliably separate XAFS signal and background with as little input from the user as possible. I also thank Daniel Haskel, Maoxu Qian, John Rehr, Bruce Ravel, Steve Zabinsky and Yanjun Zhang for many useful suggestions.

Matthew Newville
GSECARS, Bldg 434A
Argonne National Lab
Argonne, IL 60439

e-mail: newville@cars.uchicago.edu
www: http://cars.uchicago.edu/~newville/
voice: (630) 252-0431
fax: (630) 252-0443

AUTOBK version 2.63

March, 1997

Copyright © 1997 Matthew Newville

Contents

1 Introduction **1**

2 Input and Output Files **3**
2.1 Input Files 3
2.2 Output Files 3
2.3 Data File Formats 3

3 AUTOBK.INP **5**
3.1 General Format of AUTOBK.INP 5
3.2 Keywords 5
3.2.1 General and Miscellaneous Keywords 5
3.2.2 Data Input and Output Keywords 5
3.2.3 Energy Origin, Pre-Edge, and Normalization Keywords 7
3.2.4 Fourier Transform Parameters and Fitting Ranges 7

4 Energy Origin, Pre-Edge, and Normalization **10**

5 Post-Edge Background Function **11**
5.1 Overview 11
5.2 Information theory and XAFS 11
5.3 Using Splines to Approximate $\mu_0(E)$ 12
5.4 The Optimization of $\mu_0(E)$ 13

6 Examples **15**
6.1 Pure Cu example 15
6.2 Further Examples 17
6.3 Suggestions 18

List of Figures

1 Background $\mu_0(E)$ for pure Cu found by AUTOBK 16
2 Resulting $\chi(k)$ for pure Cu found by AUTOBK 17

List of Tables

1 Table 1 4
2 Table 2 9
3 Table 3 18

1 Introduction

AUTOBK[1] removes the background from x-ray absorption data in a reliable and reasonably easy-to-use manner. The XAFS is formed using the relation

$$\chi(E) = \frac{[\mu(E) - \mu_0(E)]}{\Delta\mu_0(E_0)}, \tag{1}$$

where E_0 is the absorption edge energy, $\mu(E)$ is the measured absorption coefficient, $\mu_0(E)$ is the smooth, atomic-like absorption coefficient past the edge, and $\Delta\mu_0(E_0)$ is the jump in the absorption coefficient at the edge step. $\chi(E)$, a function of photo-electron energy, is converted into $\chi(k)$, a function of photo-electron momentum using $k^2 = 2m(E - E_0)/\hbar^2$. The AUTOBK requires very little prior knowledge of the system being studied to extract $\chi(k)$ from $\mu(E)$. The resulting $\chi(k)$ has the atomic-like absorption contributions removed, but retains essentially all the local structural information about the near-neighbor environment of the absorbing atom. It is then ready for a more careful analysis of the effect of the local structure on the XAFS.

The important steps of background removal can be seen from Eq. (1) to be:

1. Determine the edge energy E_0.
2. Determine the normalization constant $\Delta\mu_0(E_0)$.
3. Determine the post-edge background function $\mu_0(E)$.

Steps 1 and 2 are not difficult, and will be discussed further in chapter 4. Step 3 is the hard part. The problem is that the true atomic-like absorption (that is, the non-XAFS absorption) will have some smooth energy dependence, but nobody knows its form. The absorption of an isolated central atom isn't good enough – $\mu_0(E)$ (the so-called embedded atom absorption) is the absorption of the central atom in the electronic environment of the solid but without the scattering from the neighboring atoms. Since it is not usually possible to measure $\mu_0(E)$ directly, it is approximated by a smooth function which has some flexibility and which can be adjusted to give some sort of fit to the measured absorption data.

AUTOBK finds $\mu_0(E)$ using concepts from Fourier signal analysis to assist the fundamental physical ideas behind the separation of XAFS and background. A piecewise polynomial, or spline, is used to approximate $\mu_0(E)$. The spline is adjusted so that the low-R components of the resulting XAFS $\tilde{\chi}(R)$ (that is, after a Fourier Transform of $\chi(k)$) are optimized. This optimization is discussed in more detail in chapter 5 — the idea is to eliminate the non-structural parts of $\tilde{\chi}(R)$ at low-R. AUTOBK controls the stiffness of $\mu_0(E)$ internally, as determined by the size of the low-R "background" range. This gives a fairly clear definition of the background (as the low-R components of the absorption), and reduces the subjectivity inherent in background removal. The result is that AUTOBK will find a reasonably good background without a lot of playing around with the data. In fact, only one parameter in the program has a profound effect on $\chi(k)$, and this (the endpoint of the low-R range) has at least some physical significance.

AUTOBK is not the only method of XAFS background removal, of course. Most standard reviews [2, 3] describe background removal, and an excellent description of the "classic" method of background removal is given by Sayers and Bunker [4]. There have been a variety of approaches to improve this method. Stern, Līviņš, and Zhang [5] used the temperature-dependence of $\mu(E)$ to *measure* a few selected points of the background

$\mu_0(E)$ for pure Pb (unfortunately, this method requires temperature-dependent XAFS data for a system dominated by single scattering). A more general approach by Cook and Sayers [6] smooths the measured $\mu(E)$ to get an approximation of $\mu_0(E)$. A technique by Li, Bridges, and Brown [7] uses FEFF calculations of the XAFS χ to determine the background as a smoothed version of $\mu(E) - \chi(E)$. AUTOBK has some obvious similarities to each of these methods. Finally, attempts to calculate $\mu_0(E)$ from first principles [8] have been partially successful, and show promise for practical data analysis. Such theoretical advancements have not been included in AUTOBK.

2 Input and Output Files

2.1 Input Files

To run AUTOBK you need an input file named ***autobk.inp*** which will control the running of the program and a data file containing measured absorption data $\mu(E)$. The form and contents of ***autobk.inp*** will be further discussed in chapter 3. The file containing $\mu(E)$ will be discussed in section 2.3. An optional input data file containing a standard $\chi(k)$ can also be given. The purpose of this standard $\chi(k)$ will be discussed in chapter 5. The input data file names can be any valid filenames up to 70 characters long allowed on your system (subdirectory paths can be given on most systems) and will be given in ***autobk.inp***. In summary, there are three inputs:

1. ***autobk.inp***, the input file for the program.
2. A data file containing the $\mu(E)$ of which the the background is to be found.
3. A data file containing the standard $\chi(k)$. (optional)

2.2 Output Files

After the background is found, AUTOBK will write ***autobk.log***, containing a synopsis of the program inputs and outputs. An output data file for the $\chi(k)$ found from the background removal will also be written. There are several additional data files that can be written, each of which contains different ways of representing data in the background removal process. Many of these are of only diagnostic interest, but some (especially those representing $\mu_0(E)$ and pre-edge subtracted $\mu(E)$) are quite convenient. Each optional output file has its own keyword to select whether or not it is written. The naming conventions for the output files will be discussed in the next section.

The most useful of the optional outputs is $\mu_0(E)$, which will be written by default (*i.e.*, unless you put `bkgout = false` in ***autobk.inp***). This will give the $\mu_0(E)$ at the same energy points as the input $\mu(E)$ data. Outside the energy region used in the background removal (*e.g.*, in the pre-edge region), this function will take the same values of the input absorption data. The background can also be written to k-space using `bkgksp = true`. The data itself can be written in R-space (that is, after the Fourier transform with the same window and k-weighting on $\chi(k)$ used for the background removal) using `datrsp = true`. If a standard $\chi(k)$ is used in the background removal, it can be written to k-space using `theksp = true` and to R-space using `thersp = true`. This will contain the standard after its amplitude has been altered in the fit.

All outputs in k-space will be written between $k_{\rm min}$ and $k_{\rm max}$. Outputs in R-space will be written between 0 and 10.0Å. Since the Fourier Transform Window (which smoothes the data and reduces "ringing" in $\tilde{\chi}(R)$) used in background removal is typically much sharper than for analysis, the R-space outputs from AUTOBK are not intended for general use, but only for diagnostic checks of the background removal.

2.3 Data File Formats

As in all UWXAFS3.0[9] data analysis programs, there are two options for the format of the data files. The data may be in either a specially formatted binary file known as a UWXAFS binary file (also called an RDF file), or in a formatted ASCII column file. More information on these file formats, including the format specifications and a

Table 1: Names of Output Files generated by AUTOBK.

Contents of File	filename (ascii)	(uwxafs)	**file type**	keyword in *autobk.inp*
data $\chi(k)$	*testk.chi*	*test.chi*	**chi**	-
$\mu_0(E)$	*teste.bkg*	*test.bkg*	**xmu**	bkgout
$\mu_0(k)$	*testk.bkg*	*test.chi*	**chi**	bkgksp
standard $\chi(k)$	*testk.stn*	*test.chi*	**chi**	theksp
standard $\tilde{\chi}(R)$	*testr.stn*	*test.rsp*	**rsp**	thersp
data $\tilde{\chi}(R)$	*testr.dat*	*test.rsp*	**rsp**	datrsp

discussion of the relative merits of the two file formats can be found in the UWXAFS3.0 reference document on data files [10]. The two file handling formats can be mixed, so that the input data can be in the UWXAFS format and the output data can be in the ASCII format, or vice versa.

If the input data are in UWXAFS format, it must be in a file with file type 'XMU'. Both the file name and record key (either nkey or skey) must be specified for the input. If the input data are in ASCII format, it must contain $\mu(E)$ data for exactly one scan in column format, with the first column containing energy in eV, and the second column containing the measured absorption coefficient. Raw data from a synchrotron must be converted before giving it to AUTOBK, and as there is no common format for such data, you'll have to do this yourself. If you find this requirement a hardship, contact us and we may be able to help. The example file *cu10k.dat* is an example of an ASCII column data file that contains $\mu(E)$. See the document on data files [10] for further formatting details.

Outputs files will be written to files with names that depend on the contents of the file, the output file format (UWXAFS or ASCII), and the user-chosen name. If the output file name is specified by `out = test` in *autobk.inp*, the output files will be named according to the Table 1. Note that the suffix depends on the type of data written. For ASCII data files, the last letter of the prefix gives the abscissa of the data.

3 AUTOBK.INP

3.1 General Format of AUTOBK.INP

Commands to AUTOBK are read from the file *autobk.inp*. These input commands name the data file to use for the $\mu(E)$ data and set all user options and controls. AUTOBK uses keywords to describe and assign values to all program parameters. The use of keywords allows the input file to be read easily and the values of the program parameters to be modified easily. The keywords have fairly transparent meanings, and are assigned values with keyword "sentences" with syntax:

`keyword` ⟨delimiter⟩`value`.

The `keyword` must be one of the valid keywords listed below. The ⟨delimiter⟩is an equal sign or white space (a blank or TAB) surrounded by any number of white spaces. The `value` is provided by the user and will be interpreted as a number, a logical flag, or a character string, depending on the nature of the keyword — the list below will indicate what kind of value each keyword takes. Logical flags all have values `true` or `false` (`t` and `f` will work, too). If a keyword's `value` is a number or logical flag (but not a character string), the assigning keyword sentence can be put on the same line as other numerical and logical keyword sentences. Keywords that take character strings as their value must occur on their own line. AUTOBK does not distinguish keywords by case. But to accommodate many operating systems, it does distinguish the names of external files by case. Keyword sentences are allowed to occur in any order in the file. Internal comments can be written anywhere in *autobk.inp*.

3.2 Keywords

The following subsections give a detailed explanation for each keyword in AUTOBK. Table 2 at the end of this chapter has a list of all the keywords for all the program controls and parameters in AUTOBK with a brief description of the meaning of their values.

3.2.1 General and Miscellaneous Keywords

`%` or `!` Indicates a comment anywhere in *autobk.inp*, including end-of-line comments.

`*` or `#` Indicates a comment line in *autobk.inp* if it is the first character on the line.

`----` Tells AUTOBK to stop reading inputs for this data set, and begin background removal on this data set. AUTOBK will return to this place in *autobk.inp* after the background is found for the current data set, and will read inputs for another data set. In this way, more than one background removal can be done with a single *autobk.inp* file.

3.2.2 Data Input and Output Keywords

See chapter 2 for further details.

`Title` User-chosen title line which will be written to the output files. This must be on its own line.

`Formin` File format to use for the input data files. The choices are UWXAFS and ASCII. See chapter 2 and the UWXAFS3.0 document on data files [10] for more details. The default is for AUTOBK to find the input format itself, from the input data. This does not need to be on its own line.

`Formout` File format to use for the output data files. The choices are the same as for `Formin`, and the default is to use the format used as the input format. This does not need to be on its own line.

`Format` Sets both `Formin` and `Formout`.

`Data` Name of input data file containing $\mu(E)$. For UWXAFS format files, this file must have file type 'XMU", and either the nkey or skey must also be given, so that the syntax must be something like:
`Data = cu.xmu, 1` or `Chi = cu.xmu , TROUT`.

For ASCII input data format, only the input file name is needed. This should be on its own line. See chapter 2 and Ref. [10] for more details.

`Theory` Name of the input data file containing $\chi(k)$ for the standard. The naming conventions will be the same as for the $\mu(E)$ data file above. If the UWXAFS format is used, this must have file type "CHI". This should be on its own line. See chapter 5 for more on the use of this standard $\chi(k)$.

`Fixamp` Logical flag to prevent the amplitude of the standard $\chi(k)$ from being rescaled automatically in the fit. The default is false, so that the amplitude of the standard will be adjusted so as to match the first shell $\tilde{\chi}(R)$ of the data between $R_{\rm bkg}$ and $R_{\rm 1st}$.

`Fixe0` Logical flag to prevent E_0 from being varied in the fit. The default is false, so that E_0 of the data $\chi(k)$ will be varied.

`Out` Prefix for the output file name. See chapter 2 for more details, and an explanation of the file name suffixes. This does not need to be on its own line.

`Bkgout` Logical flag for writing output data for $\mu_0(E)$. This data will be written at exactly the same energy points as the input $\mu(E)$ data. The default is true.

`Bkgksp` Logical flag for writing out $\mu_0(k)$. The default is false.

`Theksp` Logical flag for writing out $\chi(k)$ of the standard. The default is false.

`Thersp` Logical flag for writing out $\tilde{\chi}(R)$ of the standard. The default is false.

`Datrsp` Logical flag for writing out $\tilde{\chi}(R)$ of the data. The default is false.

`Allout` Logical flag for writing all of the above outputs.

3.2.3 Energy Origin, Pre-Edge, and Normalization Keywords

See chapter 4 for further details.

`E0` Edge Energy in eV. The default value for the starting E_0 is set near the point of maximum derivative on the absorption edge. If a standard $\chi(k)$ is used, this value will be adjusted unless the flag `fixe0` is set to `true`.

`Pre1` Low energy limit of pre-edge range, over which a line is fit to help determine the normalization constant, as discussed in chapter 4. The value is relative to E_0, and the default is -200.0 eV.

`Pre1` High energy limit of pre-edge range, over which a line is fit to help determine the normalization constant, as discussed in chapter 4. The value is relative to E_0, and the default is -50.0 eV.

`Nor1` Low energy limit of post-edge range, over which a quadratic polynomial is fit to $\mu_0(E)$ to determine the normalization constant, as discussed in chapter 4. The value is relative to E_0, and the default is 100.0 eV.

`Nor2` High energy limit of post-edge range, over which a quadratic polynomial is fit to $\mu_0(E)$ to determine the normalization constant, as discussed in chapter 4. The value is relative to E_0, and the default is 300.0 eV.

`Edge` The edge step normalization constant, $\Delta\mu_0(E_0)$ in Eq. (1). Specifying this value will overwrite the default normalization constant found as discussed in chapter 4.

3.2.4 Fourier Transform Parameters and Fitting Ranges

For further information on the meaning and effect of the Fourier Transform Parameters, see the UWXAFS3.0 reference document on Fourier transforms [11].

`Rbkg` $R_{\rm bkg}$, the maximum R over which the background function $\mu_0(E)$ is fit. Note that this value is *not* corrected for any phase-shifts, and so corresponds to $\tilde{\chi}(R)$, not the absolute interatomic distance. The default in 1.0 Å.

`R1st` $R_{\rm 1st}$, the maximum R of the first shell to use in making the amplitude of the standard and data $\tilde{\chi}(R)$ equal. As for `Rbkg`, this is not corrected for any phase shifts. The default is `Rbkg` + 2.0 , Å.

`Emin` Low-E value (relative to E_0) of the region over which the background function $\mu_0(E)$ is fitted. This value will correspond exactly to `Kmin` below, and only one needs to be specified. The default is 0. eV.

`Emax` High-E value (relative to E_0) of the region over which the background function $\mu_0(E)$ is fitted. This value will correspond exactly to `Kmax` below, and only one needs to be specified. The default is the last data point.

`Kmin` Low-k value of the region over which the background function $\mu_0(E)$ is fitted. This value will correspond exactly to `Emin` above, and only one needs to be given. The default is 0. Å^{-1}.

`Kmax` High-k value of the region over which the background function $\mu_0(E)$ is fitted. This value will correspond exactly to `Emax` above, and only one needs to be given. The default is the k of the last point data.

`Kweight` k-weighting for the Fourier Transform. The default is 1.

`Dk1` Low-k Fourier Transform Window Parameter (window "sill") for the Fourier Transform. The default is 0.0.

`Dk2` High-k Fourier Transform Window Parameter (window "sill") for the Fourier Transform. The default is 0.0.

`Dk` Sets both `Dk1` and `Dk2` to the same value. The default is 0.0.

`Iwindo` Integer index to specify which of the possible Window Types to use for the Fourier Transform. The default is 0, indicating Hanning Windows. See Ref. [11] for details of XAFS Fourier transforms, and a list of the possible Window Types.

Table 2: Keywords for AUTOBK, the format of their arguments (c for character string, n for number, l for logical flag), a brief explanation of their meaning. Where appropriate, valid values are given in (parentheses) and default values are given in [square brackets].

General/Miscellaneous:

`%`/`!`	-	End-of-Line Comment: ignore everything after `%` or `!`.
`#`	-	Comment Line: ignore line if `#` is in 1st column.
`----`	-	End of Data set: stop reading inputs for this data set.

Data Input and Output:

`Title`	c	User chosen title line	[none]
`Formin`	c	File Format of input data file	(uw/ascii)
`Formout`	c	File Format of output data files	(uw/ascii)
`Format`	c	File Format of input and output data files	(uw/ascii)
`Data`	c	Name of input data file containing $\mu(E)$	[none]
`Theory`	c	Name of input data file with $\chi(k)$ for standard	[none]
`Out`	c	Name of output data file	[same as input]
`Fixamp`	l	Flag for fixing the amplitude of the standard $\chi(k)$	(T/F) [F]
`Fixe0`	l	Flag for not fitting the value of E_0	(T/F) [F]
`Bkgout`	l	Flag for writing $\mu_0(E)$ to output file	(T/F) [T]
`Bkgksp`	l	Flag for writing $\mu_0(k)$ to output file	(T/F) [F]
`Theksp`	l	Flag for writing $\chi(k)$ of standard to output file	(T/F)[F]
`Thersp`	l	Flag for writing $\tilde{\chi}(R)$ of standard to output file	(T/F)[F]
`Datrsp`	l	Flag for writing $\tilde{\chi}(R)$ of data to output file	(T/F)[F]
`Allout`	l	Flag for writing all of the above outputs	(T/F) [F]

Pre-Edge, E_0, and Normalization:

`E0`	n	Edge energy in eV.	[found]
`Pre1`	n	Low energy of pre-edge range, relative to E_0	[-200]
`Pre2`	n	High energy of pre-edge range, relative to E_0	[-50]
`Nor1`	n	Low energy of normalization range, relative to E_0	[+100]
`Nor2`	n	High energy of normalization range, relative to E_0	[+300]
`Step`	n	Edge step	[found]

Fitting Ranges and Fourier Transform Windows:

`Rbkg`	n	$R_{\rm bkg}$, maximum R value to fit the background	[1.0]
`R1st`	n	$R_{\rm 1st}$ for fit of standard to the first shell	[`Rbkg`+2.0]
`Emin`	n	Starting E for the background spline	[0.0]
`Emax`	n	Ending E for the background spline	[last data point]
`Kmin`	n	Starting k for the background spline	[0.0]
`Kmax`	n	Ending k for the background spline	[last data point]
`Kweight`	n	k-weight for Fourier Transform	[1.0]
`Dk1`	n	Low-k Fourier Transform Window "Sill"	[0.0]
`Dk2`	n	High-k Fourier Transform Window "Sill"	[0.0]
`Dk`	n	Both `Dk1` and `Dk2`	[0.0]
`Iwindo`	n	Index of Fourier Transform Window Function	[0]

4 Energy Origin, Pre-Edge, and Normalization

In addition to fitting the background function $\mu_0(E)$, the data must be properly normalized in order to construct the $\chi(E)$ of Eq. (1). Both the normalization process and the conversion from $\chi(E)$ to $\chi(k)$ require a reasonable estimate of the threshold energy E_0. Though both E_0 and the normalization constant $\Delta\mu_0(E_0)$ can be explicitly set by the user, AUTOBK can make adequate estimates of these numbers, as will be described here.

The value for E_0 found by AUTOBK will rarely be extremely poor, but it can easily be off by a few volts from where you might pick it. By default the value of E_0 will be chosen as an energy point in the edge, near where $d\mu/dE$, the derivative of $\mu(E)$, has a maximum. Numerical derivatives are not very trustworthy and the maximum might find a glitch in the data, so E_0 is chosen more safely than simply finding the value of E with the maximum $d\mu/dE$. The initial value of E_0 (either entered by the user or found by AUTOBK) will be varied if a standard $\chi(k)$ is used for the background removal, unless the logical flag `fixe0` is explicitly set to `true`. The fitting of E_0 isn't very sensitive, because the job of AUTOBK is to get $\mu_0(E)$, which doesn't depend much on the E_0. It is rare for E_0 to be adjusted more than a few volts. The fitted value of E_0 probably shouldn't be trusted very much anyway, and should probably be more carefully determined in the analysis of $\chi(k)$.

The normalization in AUTOBK is done by a single constant number, $\Delta\mu_0(E_0)$. Because energy-dependences of x-ray detectors are usually comparable in size to the energy-dependence of $\mu_0(E)$ there is little point in normalizing by a energy-dependent background. In any event, the primary energy-dependences of the detectors and $\mu_0(E)$ are not difficult to estimate (as with the UWXAFS3.0 program ATOMS), so that these corrections can be later put into the analysis. See the example of pure Cu in the FEFFIT document for how this can be done.

The constant value of $\Delta\mu_0(E_0)$ can be set in AUTOBK using the keyword `step`. If it is not given, this constant will be found by taking the difference in the extrapolation of smooth functional fits to the pre-edge total absorption $\mu(E)$ and post-edge background absorption $\mu_0(E)$ (after it is found, of course) at the threshold energy, E_0, so that

$$\Delta\mu_0(E_0) = \mu_0^+(E_0) - \mu^-(E_0). \tag{2}$$

The measured absorption below the edge step (the so-called pre-edge region) is fit to a straight line over the energy region between $[E_0 + E_{\rm pre1}, E_0 + E_{\rm pre2}]$. Both $E_{\rm pre1}$ and $E_{\rm pre2}$ can be set by the user with keywords `pre1` and `pre2`, and have default values of -200, and -50 eV, respectively. These two numbers are relative to E_0, so they should be negative numbers that are in the measured pre-edge region of the data. This fitted line is then extrapolated to E_0, giving $\mu^-(E_0)$. The values of the slope and intercept of this pre-edge line will be written to *`autobk.log`*.

The background function $\mu_0(E)$ (found as discussed in chapter 5) is fit to a quadratic polynomial in E over the energy region between $[E_0 + E_{\rm nor1}, E_0 + E_{\rm nor2}]$. Both $E_{\rm nor1}$ and $E_{\rm nor2}$ can be set by the user with keywords `nor1` and `nor2`, and have default values of 100, and 300 eV, respectively. These two numbers are relative to E_0, and should be positive numbers that are in the measured region of the data. This fitted polynomial is then extrapolated to E_0, giving $\mu_0^+(E_0)$.

5 Post-Edge Background Function

The $\mu_0(E)$ term in Eq. (1) represents the absorption due to the deep core level of an isolated absorbing atom in the solid. This so-called embedded atom absorption will differ from the absorption for a truly isolated atom because of the overlap of electron orbitals of neighboring atoms. It should still be a smoothly varying function of energy, so that it can be well-separated from the oscillatory part of the total absorption, but its actual form is poorly known. Current state-of-the-art calculations of $\mu_0(E)$ can get only qualitative agreement with experiments, and their use in analysis is not yet reliable.

The reluctantly accepted practice in XAFS analysis is to use as the background function $\mu_0(E)$ "some reasonably smooth function which, in some manner, approximates the original $\mu(E)$". All background removal techniques use this approach, and the differences between different techniques lies in how this rather qualitative criterion is interpreted. In this chapter I'll explain how AUTOBK interprets the background criterion.

The main principle of AUTOBK is to apply information theory to the separation of the background and XAFS. This allows the qualitative background criteria to be made quantitative and codified. The bearing of information theory on XAFS will be discussed in section 5.2

5.1 Overview

AUTOBK uses a piecewise polynomial, or spline, to approximate $\mu_0(E)$. The spline is chosen to optimize the R-components of $\tilde{\chi}(R)$, the Fourier Transform of $\chi(k)$, below $R_{\rm bkg}$. The stiffness of the spline [12] is controlled by the number of knots, points at which the different polynomial pieces meet, and where there can be a discontinuity in some high order derivative. The number of knots in the background spline is chosen to be the number of independent points in the low-R range of $\tilde{\chi}(R)$, between R = [0.0, $R_{\rm bkg}$]. This is simply given by the number of independent points in this region, which is

$$N_{\rm bkg} = 1 + \frac{2\Delta k R_{\rm bkg}}{\pi}. \tag{3}$$

where $R_{\rm bkg}$ is an estimate of the low-R edge of the first peak in the resulting $\tilde{\chi}(R)$, and Δk is the k-range of the data. $N_{\rm bkg}$ is the number of degrees of freedom in the data below $R_{\rm bkg}$. Based on ideas of information theory, the knots of the spline are equally spaced in k, which will minimize the spectral leakage of the background into the region above $R_{\rm bkg}$. AUTOBK uses fourth order splines (*i.e.*, cubic splines) to ensure that no more than one full oscillation of the spline can occur between knots. This means that the highest measurable R value (the so-called Nyquist critical frequency) is $R_{\rm bkg}$, and that all components of the background above $R_{\rm bkg}$ comes from spectral leakage due to the finite k-range.

5.2 Information theory and XAFS

There is a fundamental result of information theory [13] that limits number of frequencies that can be distinguished in a signal with finite time duration. This is essentially a restatement of the uncertainty relation for Fourier conjugate variables. For signal analysis, this result (often attributed to Shannon) can be stated quantitatively like this: For a signal with time duration Δt, two frequencies cannot be distinguished if they differ by less than $\delta\omega = \pi/\Delta t$. Since only a finite frequency range (or bandwidth) can be used

for any real signal, there is a finite limit on the number of distinguishable frequencies measurable in a signal. In this sense there is an upper limit on the amount of information that can be transmitted in a signal through its different frequencies. This limit is just the maximum number of independently detectable frequencies in the signal, given by

$$N \approx \frac{\Delta t \Delta \omega}{\pi}, \tag{4}$$

where Δt is the time duration of the signal and $\Delta\omega$ is the measurable frequency range. We can eliminate the approximate nature of Eq. (4) by looking at the Fourier series upon which sampling theory is based. The interpretation from this approach is that the information is taken as the values of the Fourier coefficients, which come in pairs, spaced at intervals in ω of $\pi/2\,\Delta t$. This gives the amount of information a clear interpretation and immediately leads to

$$N = \frac{\Delta t \Delta \omega}{\pi} + 1. \tag{5}$$

The +1 represents the constant term in the Fourier series expansion, and corresponds to there being one Fourier coefficient at $\omega = 0$ and pairs of coefficients at all non-zero multiples of $\pi/2\,\Delta t$.

For XAFS, the conjugate variables are k and $2R$, we sample $\chi(k)$ between $k_{\rm min}$ and $k_{\rm max}$, and want to get structural information between $R_{\rm min}$ and $R_{\rm max}$. The amount of information we can get out of an XAFS measurement is therefore given by Eq. (5) to be simply given by

$$N = \frac{2(R_{\rm max} - R_{\rm min})\,(k_{\rm max} - k_{\rm min})}{\pi} + 1, \tag{6}$$

This is valid for the whole R-range, even the low-R region where there is no structural information in the XAFS $\tilde{\chi}(R)$! (For more on this subject, see [14]). Since XAFS analysis is intended to give information about the atomic structure, and since the low-R region has none, the interpretation in AUTOBK is that

All the information below the first peak in $\tilde{\chi}(R)$ can be used to determine the background.

We then have a clear definition of how much information can be used in getting the background, given by Eq. (3). $R_{\rm bkg}$ is the only term in this definition which depends at all on the physical details of the atomic distribution of the system, and is easily interpreted as the low-R cut-off below which the data will not be analyzed for its structural content. Finally, since we're interpreting the background information as coefficients in a Fourier series, we know the information must be equally spaced in both k- and R-space.

5.3 Using Splines to Approximate $\mu_0(E)$

A spline is a piecewise polynomial, a function made up of several contiguous polynomial sections. The places where two polynomial pieces meet are called "knots". Splines are commonly used to approximate functions that are expected to be fairly smooth, but whose actual form is not completely known. They are especially easy to use because they can be made arbitrarily flexible and are very easy to calculate in terms of a small number of degrees of freedom (typically, one for each knot). At the knots, the function must be continuous in its value, but some of its derivatives might have discontinuities. Usually only it's highest non-trivial derivative is discontinuous, so that one degree of

freedom is associated with each knot. Endpoints of the spline need to be dealt with as special cases.

AUTOBK uses something very similar to standard cubic splines (it actually uses b-splines of fourth order, but the distinction is not important for the discussion here, and the functional form of the polynomial pieces is not as important as the amount of freedom at each knot). One free coefficient is associated with each of the $N_{\rm bkg}$ knots of the spline. The value of this free coefficient is optimized as discussed in the next section. Good initial values for the free coefficients of the spline turn out to be easy to get for splines by guessing that the spline goes through the $\mu(E)$ values at the knot locations. Aside from giving the initial guesses of the spline coefficients the $\mu(E)$ data is not explicitly used for evaluating the background function. The spline is not forced in any way to go through any $\mu(E)$ points, including either of the endpoints.

5.4 The Optimization of $\mu_0(E)$

The background function $\mu_0(E)$ is chosen in AUTOBK to be a spline with $N_{\rm bkg}$ free coefficients, where $N_{\rm bkg}$, given by Eq. (3), depends only on $R_{\rm bkg}$ and the k-range of the data. These free coefficients are the ordinate values for the spline at each of its knots (which are evenly k-spaced), and completely specify the spline. In this sense the $\mu_0(E)$ depends only on the $N_{\rm bkg}$ free spline coefficients, which I'll denote as the vector $\mathbf{y}$ (with $N_{\rm bkg}$ components), so that $\mu_0(E) = \mu_0(E, \mathbf{y})$. The criterion for choosing the best $\mu_0(E)$ is then reduced to finding the $N_{\rm bkg}$ components of $\mathbf{y}$ which will minimize the non-structural components of the resulting $\tilde{\chi}(R)$ below $R_{\rm bkg}$. This expression of the "smoothness" argument for $\mu_0(E)$ can easily be solved using a least-squares minimization algorithm.

The function to minimize in the least-squares sense in AUTOBK is a function of R, but really depends only on the $N_{\rm bkg}$ spline coefficients, and is

$$f(R, \mathbf{y}) = \mathrm{FT}\left[\frac{\mu(k) - \mu_0(k, \mathbf{y})}{\Delta\mu_0(E_0)} - \chi_{\rm standard}(k)\right] \qquad R \le R_{\rm bkg}, \tag{7}$$

where FT represents the XAFS Fourier transform, as described in the UWXAFS3.0 reference document on XAFS Fourier transforms [11]. Chapter 6 gives suggested values for the XAFS Fourier transform parameters to use in background removal.

The non-linear least-squares minimization of $f(R, \mathbf{y})$ is done using the Levenberg-Marquardt algorithm[15, 16]. Since $f(R, \mathbf{y})$ is complex, both its real and imaginary components are minimized. Only the region below $R_{\rm bkg}$ is used in the minimization of $f(R, \mathbf{y})$, so there is no danger of mistakenly removing "real" data above $R_{\rm bkg}$, even though $N_{\rm bkg}$ may seem like a fairly large number of knots. In this way, all information in $\mu(E)$ with R components below $R_{\rm bkg}$ are used to give $\mu_0(E)$, and none of the components above $R_{\rm bkg}$ are used.

To ensure that only the non-structural components of $\mu(E)$ are removed in the background removal, the expected spectral leakage from the first shell XAFS from a standard $\chi(k)$ should be included in the determination of $\mu_0(E)$, as is indicated by the term $\chi_{\rm standard}(k)$ in Eq. (7). If no standard $\chi(k)$ is given, $\chi_{\rm standard}(k)$ is zero, and the minimization of $f(R, \mathbf{y})$ in Eq. (7) will be equivalent to minimizing just the low-R components of $\mu(E)$.

Since only the spectral leakage (due to the finite k-range of the data) in the low-R region will be used, this standard $\chi(k)$ needs to be only a rough estimate of the first shell

XAFS. Usually either a calculation from FEFF of the expected first shell XAFS spectra or an experimentally measured standard $\chi(k)$ (one for which the background removal is trusted) can be used as the standard $\chi(k)$. If you're planning to compare different $\chi(k)$ data in later analysis, we recommend using a single standard $\chi(k)$ (either from FEFF or experiment) for all background removals. Our experience is that the resulting background is not extremely sensitive to the details of the standard $\chi(k)$, though it is important to have the right backscattering atom at roughly the right distance.

If a standard $\chi(k)$ is used in the optimization of $\mu_0(E)$, the value of E_0 and the amplitude of the standard $\chi(k)$ can be altered in the fit. The fit of E_0 is not very accurate because there isn't much information in the low-R region that depends on E_0. So E_0 rarely gets moved by more than 5 eV. The fitting of E_0 can be turned of by saying `fixe0 = true` in *autobk.inp*. The amplitude of the standard $\chi(k)$ will normally be scaled by a constant factor so that the first shell of the standard $\tilde{\chi}(R)$ is the same size as that of the data $\tilde{\chi}(R)$. This is done to make the leakage into the low-R region roughly the right size. This scale factor is chosen to make the amplitudes of the standard and data $\tilde{\chi}(R)$ over the first shell region, $R = [R_{\rm bkg} + \pi/\Delta k, R_{\rm 1st}]$ equal in size. Note that this region begins at the pair of independent points after the background R-region, so as to prevent significant correlation of background parameters with this scale factor. $R_{\rm 1st}$ can be selected using the keyword `R1st` in *autobk.inp*, and has a default value of $R_{\rm bkg}$ +2.0Å. This adjustment of the amplitude of the standard $\chi(k)$ can be turned off by putting `fixamp = true` in *autobk.inp*.

6 Examples

All the files mentioned in this chapter should have been included in your distribution of the UWXAFS3.0 programs. If you don't have these files, contact us and we'll get them to you. All files and examples use the ASCII file type and have been renamed to guard against any programs in the UWXAFS3.0 distribution from easily overwriting them.

The distributed files include *atoms-cu.inp* which, if copied to *atoms.inp*, can be run through ATOMS[17]. This will write a *feff.inp*, which can be run though FEFF[18]. The *cu-feff.chi* file used in the example below should be exactly the same as *chi.dat* generated by FEFF. Running ATOMS and FEFF and examining the outputs for this simple example is probably a worthwhile exercise.

6.1 Pure Cu example

Shown below is *auto1.inp*. This file will need to be copied *autobk.inp* to run the program. This is about as simple as AUTOBK gets.

```
%------------------%    autobk.inp     %-------------------%
title    = Cu 10K, with standard from feff
data     = cu10k.dat                    % xmu data
output   = cu.dat
standard = cu-feff.chi                  % = chi.dat from feff
%------------------% end of autobk.inp %-------------------%
```

The *autobk.log* you get from this input file should be similar to this:

```
---------------------- automatic background removal----------------------
  autobk: version  2.61   17-jan-1995
-----------------------------------------------------------------------
Cu 10K, with standard from feff
--------------------------------------
input xmu data file name and skey: cu10k.dat  ASCII
   first document line: Cu foil at 10K
--------------------------------------
input theory chi file name and skey: cu-feff.chi  ASCII
   first document line: Cu metal chi.dat from feff601.a
--------------------------------------
output chi file : cu.chi
--------------------------------------
----------fitting parameters----------
initial value of e0        =      8977.580078
final value of e0          =      8982.461914
pre-edge range             =       -50.000000     -200.000000
pre-edge line              =  -.502631E-03 * Energy +   .541864E+01
edge step                  =         2.302949
energy range               =      8977.580078    11362.469727
k range                    =          .000000       24.950001
k weight                   =         1.000000
  fourier transform window:
```

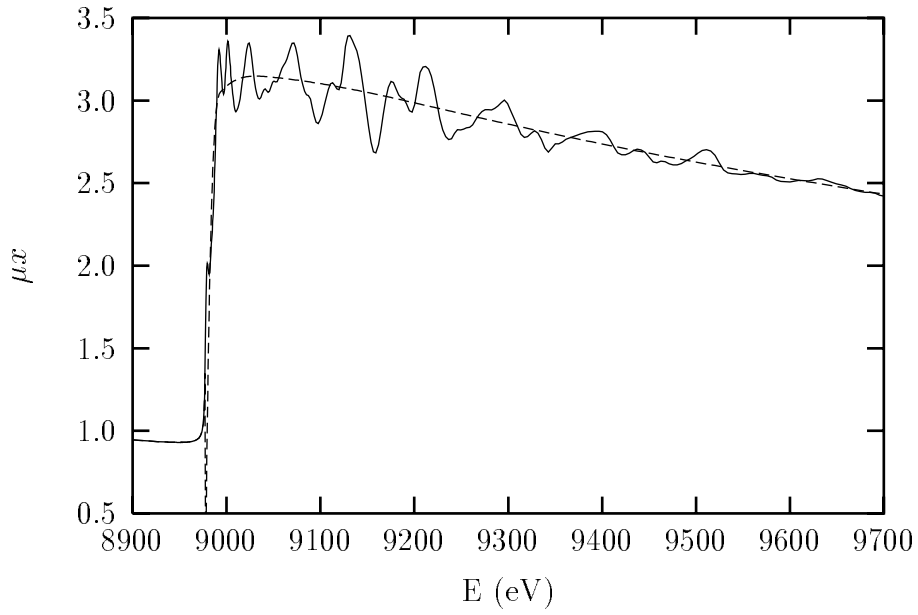


Figure 1: Measured x-ray absorption spectra for Cu at 10K (solid) and the smooth background (dashed) found by AUTOBK, using the simple input file of *auto1.inp*.

```
sills: dk1, dk2            =          .000000          .000000
# of knots in spline       =    15
background r range         =          .000000          .981748
the theory was scaled by   =          .310267
1st shell r range          =          .981748         2.945243
-------------------------------------------------------------------------
```

Don't worry too much if there are slight differences in the actual numbers you get — precisions vary for different machines. The important numbers are E_0 (which should be within a few eV of 8982eV), the number of knots in the spline, (which should be exactly 15), and the upper limit of R-range for the background, (which should be 0.98Å, the R-grid point just below 1Å, the default value for $R_{\rm bkg}$). The edge step and the slope and intercept of the pre-edge line are given (here `Energy` is the total energy, not relative to E_0, in eV). Output data files from this running of AUTOBK will be *cuk.chi*, containing the $\chi(k)$ data, and *cue.bkg*, containing the $\mu_0(E)$ data on the same grid as the input data in *cu10k.dat*.

Figure 1 shows the low energy part of the the original $\mu(E)$ and the $\mu_0(E)$ function. Note that there is a large spike in the $\mu_0(E)$ function at very low energy. Such spikes at low-energy are not unusual from AUTOBK, but do not cause much of a problem. In this case, the entire spike is below 1 Å^{-1}, which does not get very much weight in a Fourier transform into R-space, and so is not very important in the fit done by AUTOBK. Such spikes do imply that the AUTOBK method may not be optimal for getting near-edge structures, but that's another story. The resulting $\chi(k)$ data is shown in Fig. 2, which looks quite good even at low-k, the hard part of background removal. The Cu data actually extends reliably out to about 22 Å^{-1}.

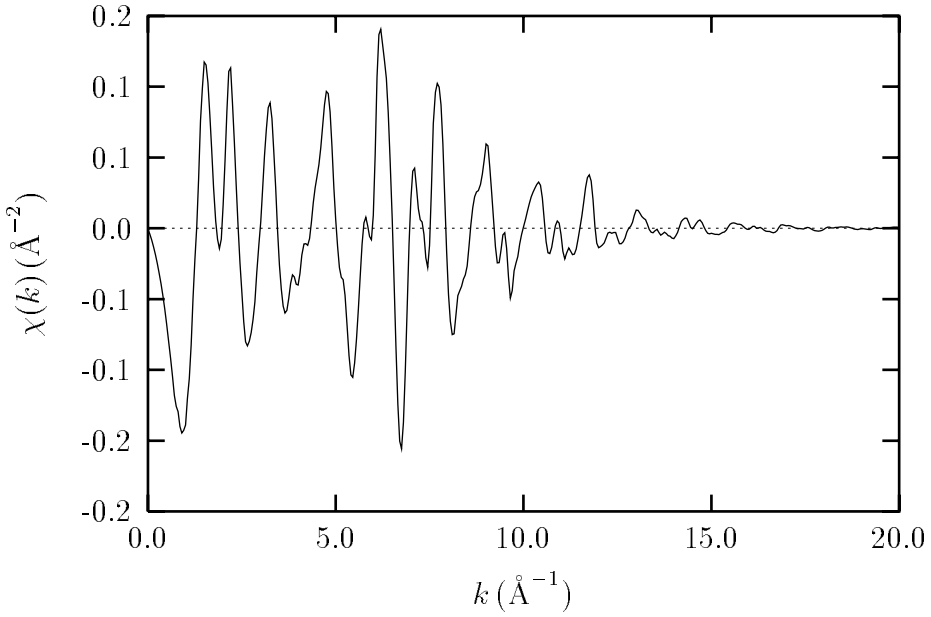


Figure 2: $\chi(k)$ for Cu at 10K (solid) found by AUTOBK, using the simple input file of *auto1.inp*

6.2 Further Examples

The above example is about all there is to running AUTOBK. You may want to add a few more of the parameters listed in chapter 3, but you really shouldn't need too many of them besides `rbkg` and `E0`. In any case, here are some more examples with the Cu data, showing how most of the important keywords are used. This one *autobk.inp* file (named *auto2.inp* in the UWXAFS3.0 distribution) actually has 3 different background removals runs in it. This is a convenient way to test the result of changing particular parameters, and for processing many scans at once.

```
%------------------%    autobk.inp     %--------------------%
title    = Cu 10K, no standard, e0 set to 8980., rbkg =1.5
data     = cu10k.dat                   % xmu data
output   = ab-1.dat
e0       = 8980.0
rbkg     = 1.5
--------------------
title    = Cu 50K, w/ standard
data     = cu50k.dat
output   = ab-2.dat
e0       = 8980.0     fixe0 = true     % don't fit e0
standard = cu-feff.chi
fixamp   = true                        % fix amp of standard
rbkg     = 1.5         kweight  = 0
--------------------
title    = Cu 50K, no standard, rbkg =1.5, kw = 1
data     = cu50k.dat                   % xmu data
```

```
output   = ab-3.dat
rbkg     = 1.5          kweight  = 1
%------------------% end of autobk.inp %--------------------%
```

Please play with the Cu data until you have a reasonably good feel for how the parameters, and decide what you like best. This should give you an idea of what kinds of inputs are needed for AUTOBK. The next section will have some more concrete suggestions.

6.3 Suggestions

Except for E_0 and $R_{\rm bkg}$, I strongly suggest using the program defaults for the numerical parameters of AUTOBK. Most importantly, do *not* use the same Fourier transform parameters that you would use for analysis (say, with FEFFIT). The hard part of background removal is the low-k (or near-edge) part of the spectrum. You want to emphasize this region, by using a small k-weighting, like 1 (the default) or maybe even 0. Also, turn the Fourier Windows "sills" off, so that `dk = 0`. Otherwise the endpoints of the k-range will get no weight in the fit and $\mu_0(E)$ will be unstable near the endpoints. Finally, start at low-k, like 0.00. If there is a strong white line you'll probably need to move $k_{\rm min}$ to just past the white line, because the AUTOBK background won't be able to follow most white lines.

The value of $R_{\rm bkg}$ is the most important parameter, and the hardest to pick. Start with 1.0 Å or half the near-neighbor distance. You may then want to adjust it to a value where $\tilde{\chi}(R)$ is a small fraction of the maximum of the $\tilde{\chi}(R)$ for the first peak. Remember that the region you use for background-removal cannot be used in analysis. In fact the analysis of the first shell should begin on the next independent data point after $R_{\rm bkg}$, namely $R_{\rm bkg} + \pi/\Delta k$, where Δk is the k-range of the data. Don't go too far into the first shell or you won't have any data to analyze! A summary of suggested ranges for the most sensitive parameters in AUTOBK is given in Table 3.

Table 3: Suggested values for a few important parameters in AUTOBK.

`Keyword`	Suggested Range of Values	Notes
`Kmin`	0.00 - 0.20	unless there is a "white line"
`Kweight`	0 or 1	3 is probably too big
`Dk`	0	not bigger than 1.
`Rbkg`	1.0 or half R_{nn}	keep $\lvert\tilde{\chi}(R_{\rm bkg})\rvert \ll \max\{\lvert\tilde{\chi}(R)\rvert\}$

References

[1] M. Newville *et al.*, Physical Review B **47**, 14126 (1993).

[2] E. A. Stern and S. M. Heald, in *Handbook of Synchrotron Radiation*, edited by E. E. Koch (North-Holland, New York, 1983), Chap. 10, pp. 995–1014.

[3] T. M. Hayes and J. B. Boyce, in *Solid State Physics*, Vol. 37 of *Solid State Physics*, edited by H. Ehrenreich, F. Seitz, and D. Turnbull (Academic Press, New York, 1982), pp. 173–351.

[4] D. E. Sayers and B. Bunker, in *X-ray Absorption: Principles, Applications, Techniques of EXAFS, SEXAFS, and XANES*, Vol. 92 of *Chemical Analysis*, edited by D. C. Koningsberger and R. Prins (John Wiley & Sons, New York, 1988), Chap. 6, p. 211.

[5] E. A. Stern, P. Līviņš, and Z. Zhang, Physical Review B **43**, 8850 (1991).

[6] J. W. C. Jr. and D. E. Sayers, Journal of Applied Physics **52**, 5924 (1981).

[7] G. Li, F. Bridges, and G. S. Brown, Physical Review Letters **68**, 1609 (1992).

[8] J. J. Rehr, C. H. Booth, F. Bridges, and S. I. Zabinsky, Physical Review B **49**, 12347 (1994).

[9] The UWXAFS3.0 analysis package is copyrighted by and available from the University of Washington. For more details, see http://krazy.phys.washington.edu/ or contact the authors.

[10] M. Newville, Data files in the UWXAFS Analysis Package, unpublished, 1994, this documentation is part of the UWXAFS3.0 package.

[11] M. Newville, Fourier Transforms in the UWXAFS Analysis Package, unpublished, 1994, this documentation is part of the UWXAFS3.0 package.

[12] C. deBoor, *A practical guide to splines* (Springer-Verlag, New York, 1978).

[13] L. Brillouin, *Science and Information Theory* (Academic Press, New York, 1962).

[14] E. A. Stern, Physical Review B **48**, 9825 (1993).

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN*, 2nd ed. (Cambridge University Press, Cambridge, 1992).

[16] P. R. Bevington, *Data Reduction and Error Analysis for the Physical Sciences* (McGraw-Hill, New York, 1969).

[17] B. Ravel, ATOMS: Crystallography and XAFS, unpublished, 1994.

[18] S. I. Zabinsky *et al.*, Physical Review B **52**, 2995 (1995).

Index

χ definition, 1
$\Delta\mu_0(E_0)$, 10
E_0, 10
k definition, 1
autobk.inp, 5
`*`, 5
`----`, 5
`Allout`, 6
`Bkgksp`, 6
`Bkgout`, 6
`Data`, 6
`Datrsp`, 6
`Dk1`, 8
`Dk2`, 8
`Dk`, 8, 18
`E0`, 7
`Edge`, 7
`Emax`, 7
`Emin`, 7
`Fixamp`, 6
`Fixe0`, 6, 10
`Format`, 6
`Formin`, 5
`Formout`, 6
`Iwindo`, 8
`Kmax`, 8
`Kmin`, 7, 18
`Kweight`, 8, 18
`Nor1`, 7, 10
`Nor2`, 7, 10
`Out`, 6
`Pre1`, 7, 10
`Pre2`, 10
`R1st`, 7
`Rbkg`, 7, 18
`Step`, 10
`Theksp`, 6
`Theory`, 6
`Thersp`, 6
`Title`, 5
`#`, 5
`%`, 5
`,` 5

Files, 3
Files,
Output, 3
Files, Data, 3
Files, Input, 3
Files, Names of Output Files, 4

Information Theory, 11
Introduction, 1

keywords, 5
knots, 11

Normalization, 10

Pre-Edge, 10

splines, 11
Suggested Values, 18