DIFFKK

Matthew Newville
Consortium for Advanced Radiation Sources
University of Chicago, Chicago, IL

Julie Olmsted Cross
Naval Research Laboratory
Washington, DC

Contents

1 Introduction

DIFFKK[1] uses x-ray absorption fine-structure (XAFS) spectra to calculate anomalous x-ray scattering factors. The input XAFS spectra can be either experimental data or a theoretical calculation (there is special support for FEFF[2] calculations), and is used to improve the bare atom anomalous scattering factors of Cromer and Liberman[3] near the absorption edge. Since XAFS is sensitive to the atomic environment of the resonant atom (through the variations of the absorption coefficient), the scattering factors from DIFFKK will also be sensitive to the local atomic structure of the resonant atom. Though scattering factors that are sensitive to chemical state and atomic environment may be useful for many x-ray scattering experiments near resonances, the primary application will probably be for the interpretation of diffraction anomalous fine-structure (DAFS)[4, 5] data.

DIFFKK is intended to be an easy-to-use and robust program. At this writing, both the program and this document are works in progress. Research is still being done on these algorithms, and on the analysis of DAFS data in general. If you have questions, comments, or suggestions about the procedures or algorithms, we'd love to hear them. Feel free to contact us by email:

Matthew Newville	newville@cars.uchicago.edu
Julie Olmsted Cross	jox@pnc.aps.anl.gov

Or visit the DIFFKK web page: http://cars.uchicago.edu/~newville/dafs/diffkk/ for the latest version of the source code and this document.

Acknowledgements

DIFFKK was written with helpful suggestions and encouragement from Charles Bouldin, Bruce Ravel, and John Rehr. The idea of using a difference Kramers-Krönig transform for improving anomalous x-ray scattering factors is discussed by D. H. Templeton[6], and dates back considerably longer for optical scattering. We used (and slightly modified) the code of Brennan and Cowan[7] for the bare atom anomalous scattering factors, which was in turn based on the work of Cromer and Liberman[3].

2 Running DIFFKK

DIFFKK is a fairly simple program to run, without many optional parameters. The main requirement is a file containing an x-ray absorption spectra $\mu(E)$. This can either be experimentally measured data or a calculated spectra from FEFF, as will be discussed in more detail in section 2.1. If you don't have a $\mu(E)$ spectra, or want just the bare-atom anomalous scattering factors, see Appendix A. DIFFKK executes the following steps:

1. Read the $\mu(E)$ data, and interpolate onto a evenly spaced energy grid.
2. Look up bare-atom scattering factors $f'_{\rm CL}(E)$ and $f''_{\rm CL}(E)$ from Cromer-Liberman.
3. Broaden $f'_{\rm CL}(E)$ and $f''_{\rm CL}(E)$ by convolving with a Lorentzian.
4. Match the scale and step-height of the XAFS $\mu(E)$ to the smooth $f''_{\rm CL}(E)$, creating an $f''_{\rm XAFS}(E)$, which includes the XAFS-like oscillations.
5. Do a Kramers-Krönig transform on the *difference* of $f''_{\rm XAFS}(E)$ and $f''_{\rm CL}(E)$, and use this and $f'_{\rm CL}(E)$ to reconstruct $f'(E)$ with fine-structure.
6. Write out the results: A single file with columns of E, f', f'', $f'_{\rm CL}$, and $f''_{\rm CL}$ will be written. A log file, describing the run, will also be written.

More detailed information on the algorithms can be found in chapter 3 and the literature references. This chapter discusses the mechanics of running the program.

DIFFKK looks for an input file named `diffkk.inp` to read its input parameters from when it starts. If this file is not found, DIFFKK will run interactively, prompting you for the data file and other inputs for the program. The meaning and format of these input parameters are discussed in section 2.2, and the interactive session will be is described in section 2.3.

After the calculation is complete, DIFFKK will write output data to a file named by the user – this file is discussed in section 2.4. A log file, `diffkk.log`, containing a synopsis of the input parameters and information about how the calculation was performed will also be written. A notable feature of this log file is that it can be used as an *input* file – that is, it can be renamed to `diffkk.inp` and used to produce identical output data as the original run.

2.1 Input XAFS Data

The input $\mu(E)$ data to DIFFKK can be either experimentally measured or a calculated spectra. Special support is provided for direct use of the `xmu.dat` files written by FEFF (version 7) when its `XANES` card is used. No pre-edge subtraction, normalization, or background subtraction is necessary for the data.

The data file (experimental or theoretical))needs to be a plain ASCII column file, with tabs or spaces delimiting the columns. Lines beginning with a '#' will be ignored and can be used a comment lines. One of the columns in the file must be absolute energy (not relative to the edge) in eV, and another column must be μ. You can specify which columns to use for the energy and absorption spectra with the `encol` and `mucol` keywords.

As mentioned above, there is special support for `xmu.dat` files generated by FEFF with the `XANES` card. `xmu.dat` is an ASCII file with data arranged in columns. The first column contains the absolute energy, the second column contains the energy relative to the edge, and the third column contains the photo-electron wavevector k. The fourth column contains the full μ, the fifth column contains the atomic background μ_0, and the sixth column contains the fine-structure χ. To use a *xmu.dat* file in DIFFKK, you'll want to make sure these keywords are set: `isfeff = true`, `encol= 1`, and `mucol= 4`. Actually, using `mucol= 5` might be interesting too, but I'll leave that up to you.

2.2 Input Parameters

DIFFKK has a few program parameters that can be specified during its run. Normally, these input parameters will be read from *diffkk.inp*. If running interactively, you will be prompted for these input parameters.

Table 1: **Keywords for DIFFKK**. The keyword, argument format, meaning, and default value are listed for each input parameter. The argument format is one of these: `char` for character string, `float` for floating point number, `int` for integer, and `log` for a logical string (which can take values `true` or `false`). Units of energy are in eV.

`keyword`	format	meaning	default
`title`	`char`	title/comment line	none
`out`	`char`	output data file name	*dk.out*
`xmu`	`char`	input data file name	none
`isfeff`	`log`	flag for using *xmu.dat* file from FEFF	false
`encol`	`int`	column for energy array	1
`mucol`	`int`	column for μ array	2
`iz`	`int`	atomic number for central atom	none
`e0`	`float`	edge energy, E_0	none
`egrid`	`float`	energy grid for output	1.0
`ewidth`	`float`	energy width for broadening	1.5
`epad`	`float`	energy grid for padding FEFF's $\mu(E)$ below E_0	5.0
`npad`	`int`	number of points for padding FEFF's $\mu(E)$ below E_0	20
`elow`	`float`	how far below data range to extend calculation	200.
`ehigh`	`float`	how far above data range to extend calculation	500.
`end`	–	end reading of *diffkk.inp*	none

All input parameters in *diffkk.inp* exist as keyword/value pairs. That is, you specify the parameter with syntax like `xmu = mydata.xmu`, with a keyword on the left, an equal sign, and then the appropriate value for the corresponding program parameter. The program parameters can can be put in any order in *diffkk.inp*. Since many parameters read only a single word for the value, several can be put on a single line, with syntax like

```
encol = 1, mucol = 2.
```

The comma is optional (and ignored), but tends to improve readability. Comments can be put anywhere in *diffkk.inp* after a '%' or '#' character, or after a line in the file that contains only the word `end`. All keywords for the DIFFKK parameters are listed in Table 1. Example input files are shown in chapter 4.

2.3 Running DIFFKK interactively

If DIFFKK cannot find the file *diffkk.inp*, it runs interactively, prompting you for each of the input parameters listed in Table 1. Reasonable guesses will be made for parameters such as `isfeff`, `encol`, `mucol`, `e0`, and `iz`, but you may override these guessed values. An interactive session would look like this:

```
--  diffkk version 1.10  --
diffkk.inp not found --  running interactively
Reading xmu data:
** name for input xmu data [xmu.dat] >
** is this a feff xmu.dat file? [y] >

What columns are energy and mu(E) in?:
(Feff puts mu in column 4, mu0 in column 5)
** column for energy [   1] >
** column for mu [   4] >

Feff's xmu.dat may need "padding" at low energies:
** number of points to pad with [  20] >
** Padding energy grid [   5.0000] >
** atomic number [  29] >
** E0 (in eV) [   8984.7700] >

mu(E) needs to be put an even energy grid:
** energy grid spacing (in eV) [   1.0000] >

For smoother results, mu(E) can be
    extrapolated past it's input data range:
 (elow  = how far below data range to extrapolate)
 (ehigh = how far above data range to extrapolate)
** elow  (in eV) [   200.0000] >
** ehigh (in eV) [   500.0000] >

Setting up broadening of Cromer-Liberman data:
** ewidth (in eV) [   1.500] >

Looking up  Cromer-Liberman f' and f''
Broadening  Cromer-Liberman f' and f''
Matching xmu data to Cromer-Liberman f''
Doing difference Kramers-Kronig transform

Ready to write out data file:
** output file name [dk.out] >

Writing output data to  dk.out
Writing summary to diffkk.log
-- diffkk done --
```

The lines beginning with `**` expect you to either type in a value or accept the default value (shown between the brackets) by hitting return. The remaining information is either to help you decide what values to use, or progress reports on what DIFFKK is doing.

2.4 Output Files

DIFFKK writes out *diffkk.log*, a brief log file containing a list of the input parameters used by the program. Also listed are the values of an E_0 shift, and the coefficients a_0, a_1, a_2, and a_3 fitted to make the $f'(E)$ derived from the input XAFS data match the Cromer-Liberman $f''(E)$ far from the absorption edge. See chapter 3 for more information about these coefficients.

Since the *diffkk.log* written after an interactive session can also be used as a *diffkk.inp* file, it might be useful to run DIFFKK interactively once when first starting each new problem, copying *diffkk.log* to *diffkk.inp*, and then editing that to fully optimize the conversion of $\mu(E)$ to $f'(E)$ and $f''(E)$.

And of course, an output data file is also written. This file is named by the `out` parameter, and is a plain ASCII file, with text lines (all beginning with a '#' character) at the top, and then 5 columns of numbers containing E, f', f'', $f'_{\rm CL}$, and $f''_{\rm CL}$.

3 Algorithms and Procedures of DIFFKK

This chapter gives justification for and detailed description of the algorithms used in DIFFKK. Since DIFFKK is likely to be most useful for DAFS analysis, many users will likely come from an XAFS background and may not be familiar with all the details of anomalous x-ray scattering. So we'll start with a quick review of the subject. For a more complete description, consult the standard x-ray references[8, 9], or one of the more recent review articles.[6] Then we'll move on to what DIFFKK does, trying to give enough detail so you can use it with some confidence.

3.1 Overview of Anomalous Scattering

The scattering power of x-rays by an atom is described by the atomic form factor f.[8, 9] The simplest approximation to the form factor is to ignore the electronic levels of the atom, and consider the scattering from a collection of free electrons. This gives

$$f_0(\mathbf{q}) = \int \rho_e(\mathbf{r}) \exp(2\pi i \mathbf{q} \cdot \mathbf{r}) d\mathbf{r}, \tag{1}$$

where $\rho_e(\mathbf{r})$ is the electron density at position $\mathbf{r}$ and $\mathbf{q}$ is the wavevector transfer of the diffracted x-ray. $f_0(\mathbf{q})$, the Thomson scattering factor, has a smooth $\mathbf{q}$-dependence and no energy dependence. In fact, calculations of $f_0(\mathbf{q})$ usually make the further approximation that the electron density is spherically symmetric. And yet, though completely neglecting the quantized electronic levels of the atom, the Thomson scattering factor is good enough to use for the analysis of most x-ray diffraction data.

Near the excitation energies of the electronic levels of an atom, however, the approximation of ignoring the electronic structure of the atom puts the Thomson scattering factor in error by $\sim$30% or so. The discrepancy is known as the "resonant scattering" or "anomalous scattering"[1]. To account for this additional scattering term, the atomic form factor f is typically written as

$$f = f_0 + f' + if'' = f_0 + \Delta f. \tag{2}$$

The resonant scattering terms f' and f'' depend on the x-ray energy E as well as $\mathbf{q}$. In the forward scattering limit (that is, when $\mathbf{q} = 0$), f'' is related to the absorption coefficient through the optical theorem,

$$\mu(E) = (4\pi N \hbar e^2 / mE) f''(E). \tag{3}$$

where N is the atomic number density, e is the electronic charge, and m the electronic mass.

Since propagation of light through matter is a causal phenomenon, $f'(E)$ and $f''(E)$ obey dispersion relations, much like those relating the real and imaginary parts of the index of refraction. Ignoring any $\mathbf{q}$-dependence of $f'(E)$ and $f''(E)$, the dispersion relations are succinctly written using the Kramers-Krönig transforms[10] as

$$f'(E) = \frac{2}{\pi} \mathrm{P} \int_0^\infty \frac{E' f''(E')}{E^2 - E'^2} dE', \tag{4}$$

$$f''(E) = -\frac{2}{\pi} \mathrm{P} \int_0^\infty \frac{f'(E')}{E^2 - E'^2} dE'. \tag{5}$$

[1] Strictly speaking, anomalous scattering occurs when $dn/d\omega < 0$ for index of refraction n and angular frequency ω. In modern usage, this distinction is often ignored, and anomalous and resonant scattering have taken on the same meaning – scattering which is influenced by core electronic levels. We'll try to be accurate without being pedantic.

Here "P" indicates the principle value of the integral at the singularity. The major point for us is that we can determine $f'(E)$ if we know $f''(E)$ (which can be determined from $\mu(E)$), and assume that the $\mathbf{q}$-dependence of $f'(E)$ and $f''(E)$ can be safely ignored.

3.2 Using $\mu(E)$ to improve $f'(E)$ and $f''(E)$

Until the recognition that DAFS can provide more information about crystals than is available from either x-ray diffraction or absorption alone, [4, 5] anomalous x-ray scattering measurements were typically constructed to avoid the energy-dependent oscillations in the diffracted intensity just above the resonant energies of bound atomic electronic levels. For energies just below the absorption threshold, XAFS-like oscillations are not needed to accurately model $f'(E)$ or $f''(E)$. Instead, these functions can, and have been, calculated and tabulated for isolated atoms.[3] These bare-atom values for $f'(E)$ and $f''(E)$ appear to work well for modeling anomalous x-ray diffraction data, including the popular MAD phasing technique for macromolecules.

But to model DAFS data, accurate values for $f'(E)$ and $f''(E)$ are needed at *and above* the absorption threshold energy, in the standard XAFS energy range. The bare-atom values for $f'(E)$ and $f''(E)$ from Cromer-Liberman are not sufficient for DAFS analysis, and the fine-structure in both $f'(E)$ and $f''(E)$ due to solid-state effects are needed. Ideally, both experimental and theoretical values for $\Delta f(E)$ could be used to model DAFS data. Unfortunately, $f'(E)$ and $f''(E)$ are more difficult to measure directly than $\mu(E)$. And for historical reasons, theoretical understanding and reliable calculations for the fine-structure in $\mu(E)$ have been developed before the fine-structure calculations for $\Delta f(E)$. For these reasons, DIFFKK uses the simple relation between $\mu(E)$ and $f''(E)$ in Eq. 3 and the Kramers-Krönig transform of Eq. 4 to give $f'(E)$ and $f''(E)$ with fine-structure.

Of course, we cannot ignore the contributions from off-resonant electronic levels, either of the resonant atom or even some other atom in the material. Nor can we ignore the importance of the absolute scale of Δf. These two aspects of x-ray scattering are typically ignored when doing XAFS, but they cannot be ignored for DAFS data. So the basic scheme of DIFFKK is to use the bare-atom calculations from Cromer and Liberman[3] for Δf, which are assumed to be valid far from resonance, and to use $\mu(E)$ data to "put the wiggles on" the Cromer-Liberman calculation. For convenience and efficiency, DIFFKK uses the computer algorithms of Brennan and Cowan[7] (slightly modified to be more compact and portable), for the bare-atom calculation of Δf.

3.3 Cromer-Liberman values for anomalous x-ray scattering

DIFFKK generates the Cromer-Liberman functions $f'_{\rm CL}(E)$ and $f''_{\rm CL}(E)$ for the bare resonant atom. The atomic number of the resonant atom used to generate these values is set with the `iz` keyword in *`diffkk.inp`*. Data for $f'_{\rm CL}(E)$ and $f''_{\rm CL}(E)$ are determined on a uniform energy grid specified by the `egrid` parameter, and over the energy range specified by the input data range and the `elow` and `ehigh` parameters.

DIFFKK then broadens the Cromer-Liberman data by convolution with a Lorentzian whose width is set by the `ewidth` parameter. Though this has essentially no effect on $f'_{\rm CL}(E)$ and $f''_{\rm CL}(E)$ far from resonance, it does smooth out the very sharp step – 1 data point wide – in

$f''_{\rm CL}(E)$ right at the resonant energy. The broadening compensates for the finite energy resolution of the measurement and the finite width of the resonance due to the finite lifetime of the excited core hole, both of which are neglected in the Cromer-Liberman calculation.

3.4 Matching $\mu(E)$ to Cromer-Liberman $f''(E)$

After obtaining broadened $f'_{\rm CL}(E)$ and $f''_{\rm CL}(E)$ functions, DIFFKK converts the XAFS $\mu(E)$ into $f''(E)$ by matching to $f''_{\rm CL}(E)$. XAFS data is typically *not* measured as the full absorption cross-section, with the relative values of the pre-edge and post-edge $\mu(E)$ in the proportions set by the full absorption cross-sections. In fact, the pre-edge cross-section is usually deliberately suppressed, and the edge-step is usually normalized to 1. So we set $f''(E)$ to

$$f''(E) = a_0 + a_1\mu(E')(E'/E_0) + (E' - E_0)a_2 + (E' - E_0)^2 a_3. \tag{6}$$

Here, $E' = E + \Delta E$, so as to allow a shift of the energy scales between the Cromer-Liberman values and the input $\mu(E)$ data, and E_0 is the nominal value of the resonant energy (set with the `e0` keyword). The values of a_0, a_1, a_2, a_3, and ΔE are all adjusted until this $f''(E)$ best matches $f''_{\rm CL}(E)$ (in the least-squares sense) between $E_0 - E_{\rm low}$ and $E_0 + E_{\rm high}$.

In addition, there is some special processing needed for data from FEFF. This is because FEFF calculates only the *resonant* part of $\mu(E)$, and does not calculate $\mu(E)$ below the absorption edge. Since there is essentially no data below the edge, this makes it a bit hard to carefully match the step height of the calculation to the Cromer-Liberman $f''(E)$. So to compensate for this shortcoming, FEFF's $\mu(E)$ is "padded" with zeros in the pre-edge region. The two relevant program parameters `npad` and `epad` give the number of points to add, and their energy spacing, respectively. It seems that `npad=5` and `epad = 20` are usually adequate.

3.5 Numerical Kramers-Krönig Transforms

The Kramers-Krönig transform of Eq. 4 is not extremely useful for numerical calculations using real data. The main problem is that the integrals extend from 0 to ∞, which is an inconvenient energy range over which to measure $f''(E)$. And since dispersions relations like this integral are not "local in energy" (that is, $f'(E_1)$ depends on values of $f''(E)$ very far $E = E_1$), we can not simply cut the integral to some easily measured range and make a correction to get back the correct answer Fortunately, the Kramers-Krönig transform is *linear*, so we can break $f'(E)$ and $f''(E)$ into pieces which are themselves Kramers-Krönig conjugate pairs. Since $f'_{\rm CL}(E)$ and $f''_{\rm CL}(E)$ already satisfy this requirement[2], this gives a convenient way to break up the integral. We have

$$f'(E) = f'_{\rm CL}(E) + \frac{2}{\pi}\,\mathrm{P}\int_0^\infty \frac{E'[f''(E') - f''_{\rm CL}(E')]}{E^2 - E'^2}dE'. \tag{7}$$

The integral here is the *difference* Kramers-Krönig transform. It is easier to calculate numerically than the full Kramers-Krönig transform of Eq. 4 because $[f''(E') - f''_{\rm CL}(E')]$ is only non-zero within $\sim$1000eV of the resonant energy. This is the crux of the whole method: The

[2] Well, to within the accuracy needed. The Cromer-Liberman calculation do break down at very high energies, but assuming that you've kept your x-rays below 100 KeV, the effects of pair-production can probably be neglected, and then the Cromer-Liberman calculations are causal *enough*. See Ref [11] for more information.

Kramers-Krönig transform of the part of $f''(E)$ due to the local environment gives the part of $f'(E)$ due to the local environment.

The difference Kramers-Krönig integral can be done over a finite energy range, but it is still a singular integral, so there is some question about the best way to do it numerically. For that, we rely on Ohta and Ishida[12], who compared several numerical methods for Kramers-Krönig transforms, and recommended using a MacLaurin series approach for both speed and accuracy. In this sense, the integral in Eq. 7 can be approximated at each energy value E_i as

$$f'(E_i) = f'_{\mathrm{CL}}(E_i) + \frac{4(E_N - E_1)E_i}{\pi(N-1)} \sum_{k=1}^{N/2} \frac{[f''(E_k) - f''_{\mathrm{CL}}(E_k)]}{E'^2 - E^2} \tag{8}$$

where $k = 2j - i^{-1}$. In essence, this is a simple Simpson's rule integration with the sum being over every-other point – to get $f'(E_i)$ for even i, sum over the *odd* indices j of $f''(E_j)$. This conveniently avoids the division by zero at $i = j$, and still provides enough data to be accurate. It does, however, lead to a few potential problems to watch out for. First, the $f''(E)$ data must be sampled at a high enough frequency so that using every other point is sufficiently accurate. Fortunately, the rather slow decay of $1/(E'^2 - E^2)$ helps assure that enough points of $f''(E)$ actually count significantly in $f'(E)$ that this shouldn't be a serious concern for data sampled at 1eV or so. The second problem is that an abrupt step in $f''(E) - f''_{\mathrm{CL}}(E)$, which can happen if E_0 of the input $\mu(E)$ is not exactly at the Cromer-Liberman E_0 for that resonance, can cause "odd-even noise" where the resulting $f'(E)$ has some small non-smooth variations between adjacent points. This can usually be reduced, if not eliminated, by broadening the Cromer-Liberman $f''(E)$ and $f'(E)$, which effectively spreads the discontinuity in the edge-step enough so that it is seen equivalently by both the even and odd indices, and by making sure that the bare atom calculation and input $\mu(E)$ agree on E_0 for the resonance.

4 Examples

Now for a couple examples. Both use Cu as the resonant atom. The first uses measured absorption data for Cu metal, which isn't all that interesting a problem for DAFS, but illustrates the basics of using DIFFKK. The second example uses a *xmu.dat* file from FEFF for $YBa_2Cu_3O_{6.68}$, which is a problem slightly more appropriate for DAFS. The files mentioned in this chapter are all avaiable from the DIFFKK web page:

http://cars.uchicago.edu/~newville/dafs/diffkk/

4.1 Example 1: Experimental Data for Metallic Cu

Here is a *diffkk.inp* for experimental $\mu(E)$ data of metallic Cu. This file (which is called *dk-exp.inp* in the distribution), was derived from running DIFFKK interactively.:

```
%------------------------%
   title = Cu XAFS data
     out = exp.fpp       % output file name
     xmu = cu_exp.xmu    % xmu data file name
  isfeff = false         % is this a feff xmu.dat?
   encol =    1          % energy column
   mucol =    2          % mu(E)  column
      iz =   29          % Z  of central atom
      e0 =   8977.9700   % edge energy
   egrid =      1.0000   % energy grid
  ewidth =      1.5000   % for broadening CL
    elow =    200.0000   % \ how far below and above
   ehigh =    500.0000   % / the data range to go
%------------------------%
```

Running DIFFKK with this input file produces *exp.fpp* as the output file. Figure 1 shows the $f''(E)$ derived from the XAFS $\mu(E)$, scaled to match the Cromer-Liberman function far from resonance. That would be "the input" to DIFFKK. Figure 2 shows "the output" – $f'(E)$ from the difference Kramers-Krönig transform of $f''(E)$. the bare-atom $f''(E)$ and $f'(E)$ from Cromer-Liberman are shown for comparison.

4.2 Example 2: using *xmu.dat* from FEFF

Now we'll use a *xmu.dat* file from FEFF with DIFFKK to generate theoretical $f'(E)$ and $f''(E)$. Since FEFF calculates the fine-structure of a single atom in a cluster, it can generate the spectra of an atom at any crystallographic site. XAFS measurements give the average over all atoms in the unit cell. So, in order to model the DAFS for an atom that occupies more than one site in a unit cell, using experimental XAFS data will not be good enough, and getting $f''(E)$ and $f'(E)$ from FEFF[3] is the only way.

[3]Well, I suppose you *could* use some other theoretical XAFS calculation.

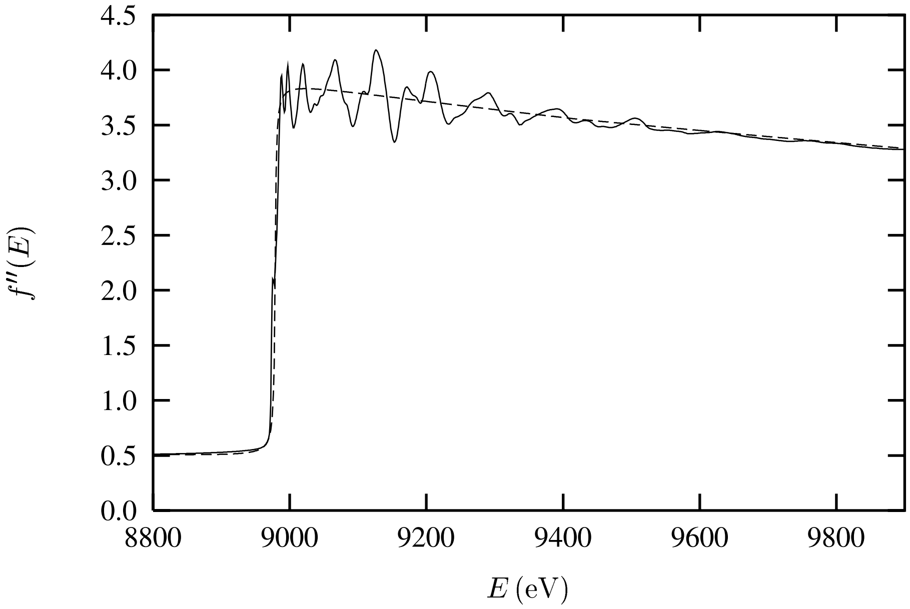

Figure 1: $f''(E)$ for pure Cu. Shown are the bare-atom $f''(E)$ from Cromer-Liberman (dashed) and the $f''(E)$ derived from the XAFS $\mu(E)$ (solid), scaled to match the Cromer-Liberman $f''(E)$ far from resonance.

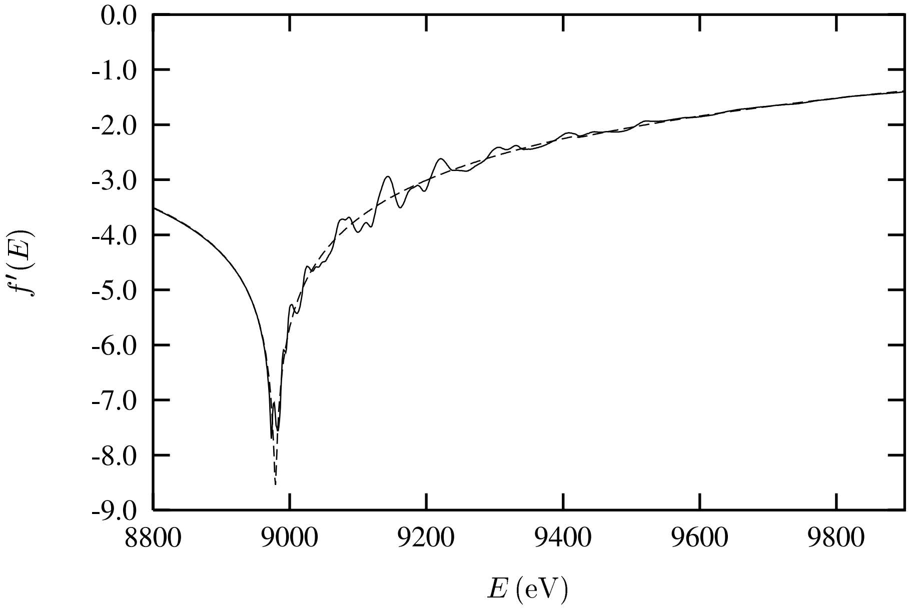

Figure 2: $f'(E)$ for pure Cu. Shown are the the bare-atom $f'(E)$ from Cromer-Liberman (dashed) and the $f'(E)$ resulting from the difference Kramers-Krönig transform (solid) using the two $f''(E)$ curves from Figure 1.

The FEFF calculation here is for the Cu(1) site in $YBa_2Cu_3O_{6.68}$. It's not crucial for the demonstration here, but the calculation included 78 scattering paths out to 7Å. More importantly for this application, FEFF7 needed the XANES card set for it to generate the full $\mu(E)$ (otherwise it calculates the fine-structure χ without the atomic-like background). The *feff.inp* used to generate this *xmu.dat* is available at the DIFFKK web site.

To generate $f'(E)$ and $f''(E)$, this *diffkk.inp* (which is called *dk-feff.inp* in the distribution) was used:

```
%------------------------%
  title  = Cu(1) of YBCO
     out = feff.fpp       % output file name
     xmu = xmu.dat        % xmu file name
  isfeff = true           % is the a feff xmu.dat?
   encol =    1           % energy column
   mucol =    4           % mu(E) column
      iz =   29           % Z of central atom
      e0 =   8986.0700    % edge energy
   egrid =      1.0000    % energy grid
  ewidth =      1.5000    % for broadening CL data
    elow =    200.0000    % \ how far below and above
   ehigh =    500.0000    % / the data range to go
    epad =      5.0000    % \ energy grid & # of points
    npad =   20           % / for pre-padding feff mu(E)
%------------------------%
```

Running DIFFKK with these settings produced the file *feff.fpp*. Figure 3 shows $f''(E)$ derived from FEFF's $\mu(E)$, and Figure 4 shows the $f'(E)$ from the difference Kramers-Krönig transform of $f''(E)$. The bare-atom $f''(E)$ and $f'(E)$ from Cromer-Liberman are shown for comparison.

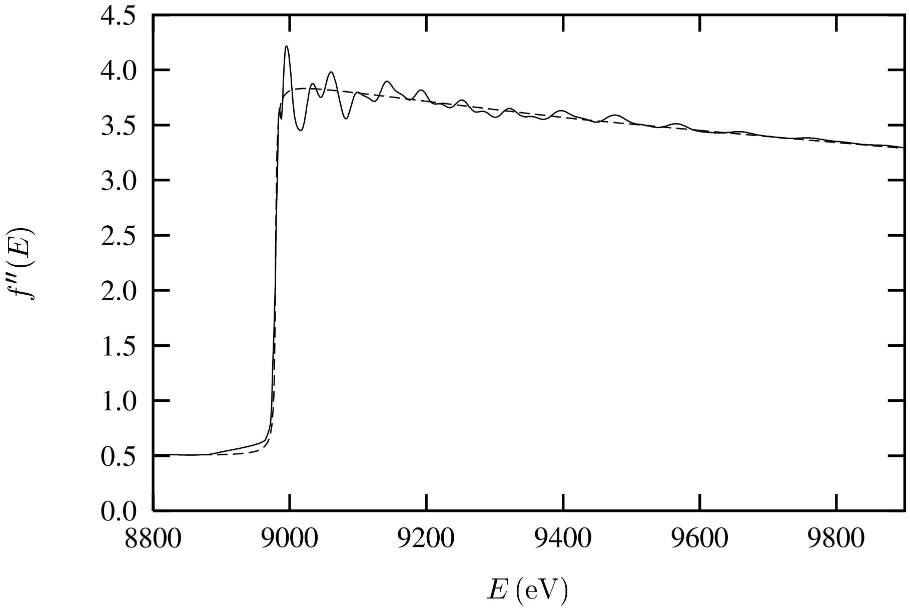

Figure 3: $f''(E)$ for the Cu(1) site of $\mathrm{YBa_2Cu_3O_{6.68}}$. Shown are the bare-atom $f''(E)$ from Cromer-Liberman (dashed) and the $f''(E)$ derived from the $\mu(E)$ calculated by FEFF (solid). As for the experimental data, the FEFF calculation was scaled to match the Cromer-Liberman $f''(E)$ far from resonance.

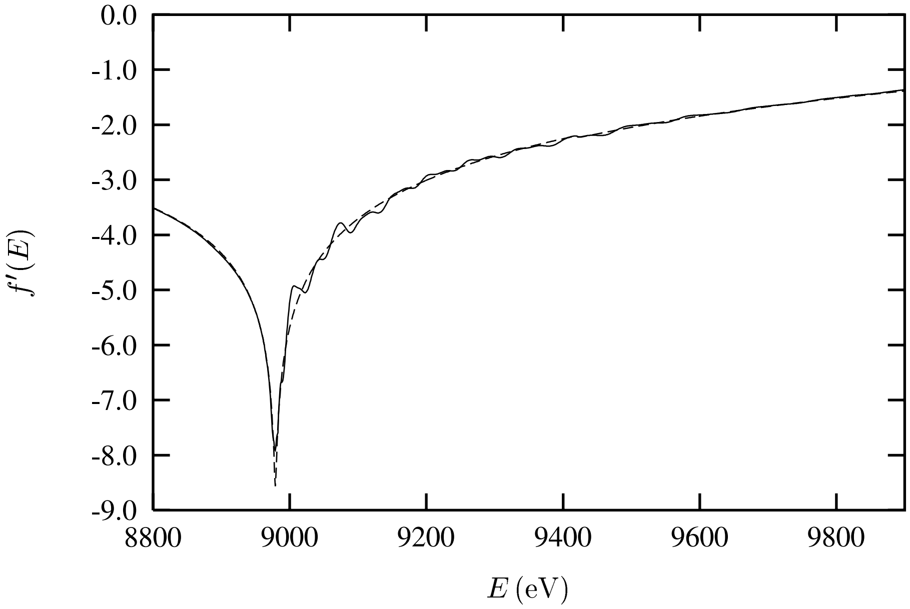

Figure 4: $f'(E)$ for the Cu(1) site of $\mathrm{YBa_2Cu_3O_{6.68}}$. Shown are the bare-atom $f'(E)$ from Cromer-Liberman (dashed) and the $f'(E)$ from the difference Kramers-Krönig transform (solid).

A DELTAF: bare atom resonant scattering factors

Though the main purpose of DIFFKK is to "put the XAFS wiggles" on the bare-atom anomalous scattering factors, those bare-atom scattering factors are quite useful on their own. Even for DAFS analysis, it can be very important to include the anomalous scattering factors from non-resonant atoms in order to fully model the intensity. Because of these needs (and because it was easy to do), DIFFKK is distributed with the companion program DELTAF to generate the bare-atom anomalous scattering factors. This calculation is based completely on the Brennan-Cowan[7] implementation of the Cromer-Liberman[3] work. There are, of course, many simliar programs available.

DELTAF runs in a similar fashion to DIFFKK. First it looks for a file `deltaf.inp`. If it finds this file it reads it and follows the instructions given there. If it can't find `deltaf.inp`, DELTAF runs interactively, prompting you for the inputs it needs. In either case, `deltaf.log` will be written, showing the program parameters used. And `deltaf.log` is in a valid format for `deltaf.inp`. Due to its simpler nature, DELTAF has even fewer parameters than DIFFKK. A full list is given in Table 2.

Table 2: **Keywords for DELTAF**. The keyword, argument format, meaning, and default value are listed for each input parameter. The argument format is one of these: `char` for character string, `float` for floating point number, `int` for integer, and `log` for a logical string (which can take values `true` or `false`). Units of energy are in eV.

`keyword`	format	meaning	default
`title`	`char`	title/comment line	none
`out`	`char`	output data file name	*`df.out`*
`iz`	`int`	atomic number for central atom	8
`elow`	`float`	lower value for energy range	3000
`ehigh`	`float`	upper value of energy range	5000
`egrid`	`float`	energy grid for output	1
`ewidth`	`float`	energy width for broadening	1.50
`end`	–	end reading of *`deltaf.inp`*	none

As you can probably tell from this table, DELTAF calculates the $f'(E)$ and $f''(E)$ from the Cromer-Liberman tables in a range [`elow`,`ehigh`] on an energy grid of `egrid`. It will apply a Lorentzian broadening factor defined by `ewidth`.

B Installation

DIFFKK is written in fortran, and should build and install without difficulty on all systems with a fortran compiler (or a C compiler and f2c). Since DIFFKK is fairly new, and still uder some development, it is probably best to get the latest source code from the DIFFKK web page: http://cars.uchicago.edu/˜newville/dafs/diffkk/. Binary executables for Windows95/NT and Macintosh systems will be made available from the web page as well, but may not be kept as up-to-date as the source. Contact Matt if you need one of these binary versions.

The DIFFKK source is available as a single (enormous) file or as compressed archive that will uncompress into the code broken apart into subroutines with a Unix-style Makefile. Unix users should probably get the compressed archive, *`difkk.tar.gz`*, and then follow the "usual" Unix installation:

Unpack the source code: Type `gzip -dc diffkk.tar.gz | tar xvf -`. This creates a subdirectory diffkk_src, which you should change into (probably by typing `cd diffkk_src`).

Customize the Makefile: This shouldn't be necessary, but you may want to edit the Makefile to change the "F77" variable (which tells how to run the fortran compiler and is set to "f77 -O1" by default), and the "INSTALLDIR" variable (which tells where to copy the files to if you type `make install`, and is set to /usr/local/bin/ by default.)

Build the programs: Type `make`. This will create the programs *`diffkk`* and *`deltaf`*.

Compilation of the single-source files should be easier, generally something like `f77 -o diffkk diffkk.f`, but a little more painful to deal with if anything goes wrong. In either case, the compilation of the code for the Cromer-Liberman tables will be fairly slow.

If you have any difficulties compiling or questions about the source code, contact Matt.

References

[1] J. O. Cross *et al.*, Phys. Rev. B **accepted**, 0 (1998).

[2] S. I. Zabinsky *et al.*, Phys. Rev. B **52**, 2995 (1995). See http://leonardo.phys.washington.edu/feff/.

[3] D. T. Cromer and D. Liberman, J. Chem. Phys. **53**, 1891 (1970).

[4] L. B. Sorensen *et al.*, in *Resonant Anomalous X-Ray Scattering: Theory and Applications*, edited by G. Materlik, C. J. Sparks, and K. Fischer (North-Holland, Amsterdam, 1994), pp. 389–420. See http://cars.uchicago.edu/˜newville/dafs/icas/.

[5] H. Stragier *et al.*, Phys. Rev. Lett. **69**, 3064 (1992).

[6] D. H. Templeton, in *Handbook of Synchrotron Radiation*, edited by G. S. Brown and D. E. Moncton (North-Holland, New York, 1991), Vol. 3, pp. 201–220.

[7] S. Brennan and P. L. Cowan, Rev. Sci. Instrum. **63**, 850 (1992). See http://www-ssrl.slac.stanford.edu/absorb.html.

[8] R. W. James, *The Optical Principles of the Diffraction of X-rays* (Ox Bow Press, Woodbridge, CT, 1962).

[9] B. E. Warren, *X-ray Diffraction* (Dover Publications, Inc, New York, 1969).

[10] J. D. Jackson, *Classical Electrodynamics* (John Wiley & Sons, New York, 1975).

[11] R. H. Pratt, L. Kissel, and P. M. Bergstrom, Jr., in *Resonant Anomalous X-Ray Scattering: Theory and Applications*, edited by G. Materlik, C. J. Sparks, and K. Fischer (North-Holland, Amsterdam, 1994), pp. 9–33.

[12] K. Ohta and H. Ishida, Applied Spectroscopy **42**, 952 (1988).

Index