EXAFS Data Analysis with FEFF

Matthew Newville

Center for Advanced Radiation Sources
The University of Chicago

July-2021

Fundamentals of X-ray Absorption Fine-Structure

Virtual XAFS School at Illinois Institute of Technology and Advanced Photon Source

The EXAFS Equation

The XAFS Equation used with FEFF:

$$\chi(k) = \sum_j \frac{S_0^2 N_j f_j(k) e^{-2R_j/\lambda(k)} e^{-2k^2\sigma_j^2}}{k{R_j}^2} \sin[2kR_j + \delta_j(k)]$$

- $f(k)$ and $\delta(k)$ are *photo-electron scattering amplitude and phase*:
  - Energy dependent $k \sim \sqrt{(E - E_0)}$.
  - Depend on Z of the scattering atom(s).
  - Non-trivial: must be calculated or carefully extracted from measured spectra.
- $\lambda(k)$ tells how far the photo-electron can travel.
- The sum is over *Scattering Paths* of the photo-electron, from absorbing atom to neighboring atom(s) and back. May include *multiple scattering*!

If we know $f(k)$, $\delta(k)$, and $\lambda(k)$, we can get:

- R – near neighbor distance.
- N – coordination number.
- σ^2 – mean-square disorder in R.

To model XAFS as a Sum of Paths:

$$\chi(k) = \sum_j \frac{S_0^2 N_j f_j(k) e^{-2R_j/\lambda(k)} e^{-2k^2\sigma_j^2}}{k{R_j}^2} \sin[2kR_j + \delta_j(k)]$$

we may refine these Parameters *For Each Path*:

XAFS Equation	LARCH Paramaeter	Physical Meaning
$S_0^2 N_j$	`s02`	Amplitude Factor: Both N_j and S_0^2
E_0	`e0`	Energy Shift (where $k = 0$)
ΔR	`deltar`	Change in path length $R_j = \Delta R_j + R_{\mathrm{eff}\,j}$
σ_j^2	`sigma2`	Mean-square-displacement in R_j

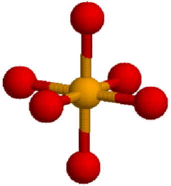
- R_{eff} is the starting R value for the FEFF Path.
- Other Parameters: higher order cumulants, energy broadening, . . .
- In principle, any parameter for any path could be refined.

EXAFS Analysis: Modeling the 1st Shell of FeO

FeO has a rock-salt structure.

To model the Fe K edge EXAFS of FeO, we'll calculate the `feffNNNN.dat` files (with $f(k)$ and $\delta(k)$), for Fe-O based on the FeO crystal structure.

We'll then *refine* the values R, N, σ^2, and E_0 so our model EXAFS function matches our data.



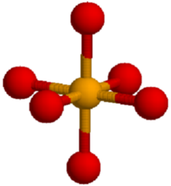
Fe-O octahedra, $R = 2.14\,\mathrm{\AA}$.

EXAFS Analysis: Modeling the 1st Shell of FeO

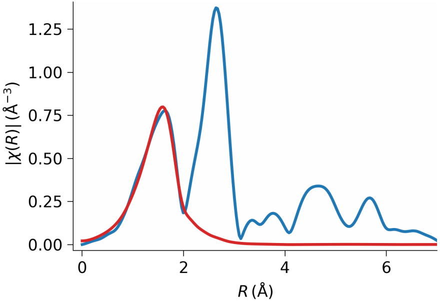
FeO has a rock-salt structure.

To model the Fe K edge EXAFS of FeO, we'll calculate the *feffNNNN.dat* files (with $f(k)$ and $\delta(k)$), for Fe-O based on the FeO crystal structure.

We'll then *refine* the values R, N, σ^2, and E_0 so our model EXAFS function matches our data.



Fe-O octahedra, $R = 2.14\,\text{Å}$.

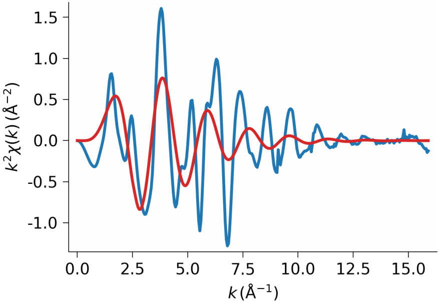


Results:

$$
\begin{aligned}
S_0^2 &= 0.7 \text{ (fixed)} \\
N &= 5.1 \pm 0.4 \\
R &= 2.09 \pm 0.01\text{Å} \\
\Delta E_0 &= \text{-}1.3 \pm 0.9 \text{ eV} \\
\sigma^2 &= 0.012 \pm 0.002 \ \text{Å}^2.
\end{aligned}
$$

$|\chi(R)|$ for FeO data and 1^{st} shell fit.

Analysis Example: 1st Shell of FeO

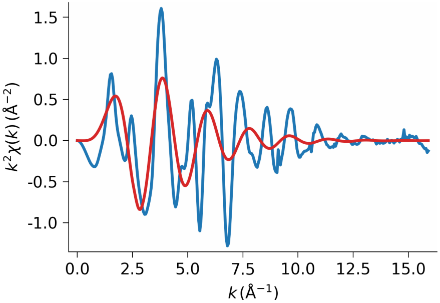


1st shell fit in k space.

Yes, that is the best fit! But only to the first shell, completely ignoring $R > 2$Å.

There is clearly another component in the XAFS besides just Fe-O.

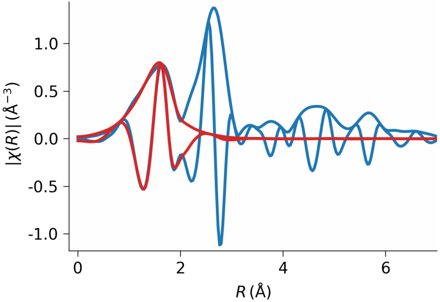
Analysis Example: 1st Shell of FeO



1^{st} shell fit in k space.

Yes, that is the best fit! But only to the first shell, completely ignoring $R > 2\text{Å}$.

There is clearly another component in the XAFS besides just Fe-O.



1^{st} shell fit in R space.

$|\chi(R)|$ and $\mathrm{Re}[\chi(\mathrm{R})]$ for FeO (blue), and a 1^{st} shell fit (red).

Although the fit to the magnitude is not perfect, the fit to $\mathrm{Re}[\chi(\mathrm{R})]$ is very good.

Fitting Strategies

Data analysis seeks a *Model* that best matches a *Measurement*.

We'll use χ^2 (don't confuse with EXAFS χ!!) to describe how good the match is:

$$\chi^2 = \sum_{i}^{N_{\rm fit}} \frac{[\chi_i^{\rm measured} - \chi_i^{\rm model}(x)]^2}{\epsilon^2}$$

where

- $N_{\rm fit}$ = number of points in the data to fit.
- ϵ = the estimated noise level in the data.
- x is the set of parameters to be varied in the analysis

The Best Fit is the one with lowest χ^2.

Fitting Strategies

Data analysis seeks a *Model* that best matches a *Measurement*.

We'll use χ^2 (don't confuse with EXAFS χ!!) to describe how good the match is:

$$\chi^2 = \sum_i^{N_{\rm fit}} \frac{[\chi_i^{\rm measured} - \chi_i^{\rm model}(x)]^2}{\epsilon^2}$$

where

- $N_{\rm fit}$ = number of points in the data to fit.
- ϵ = the estimated noise level in the data.
- x is the set of parameters to be varied in the analysis

The Best Fit is the one with lowest χ^2.

Questions:

1. How do I know how many independent measurements I have?
2. What is ϵ for my data?
3. What parameters can/should I vary?

The Information Content of EXAFS

The number of parameters we can reliably extract from our data is limited:

$$\boxed{N_{\mathrm{idp}} \approx \frac{2\Delta k \Delta R}{\pi}}$$

where Δk and ΔR are the k- and R-ranges of the usable data.

The Information Content of EXAFS

The number of parameters we can reliably extract from our data is limited:

$$\boxed{N_{\rm idp} \approx \frac{2\Delta k \Delta R}{\pi}}$$

where Δk and ΔR are the k- and R-ranges of the usable data.

For a typical range of $k = [3.0, 12.5]\,\text{Å}^{-1}$ and $R = [1.0, 3.0]\,\text{Å}$, there are ~ 12 parameters that can be determined from EXAFS. That's not much!

The Fit statistics and confidence in the measured parameters need to reflect this. But we usually oversample our data ($N_{\rm fit} > N_{\rm idp}$) so we have

$$\boxed{\chi^2 = \frac{N_{\rm idp}}{\epsilon^2 N_{\rm fit}} \sum_i^{N_{\rm fit}} [\chi_i^{\rm measured} - \chi_i^{\rm model}(x)]^2}$$

Note: I also assumed ϵ is a constant.

Other Fitting Statistics

Other "goodness-of-fit statistics":

chi-square: As before:

$$\chi^2 = \frac{N_{\rm idp}}{\epsilon^2 N_{\rm fit}} \sum_i^{N_{\rm fit}} [\chi_i^{\rm measured} - \chi_i^{\rm model}(x)]^2$$

R-factor: $\mathcal{R}$ gives a "fractional misfit" (and not scaled by the data uncertainty ϵ):

$$\mathcal{R} = \frac{\sum_i^{N_{\rm fit}} [\chi_i^{\rm measured} - \chi_i^{\rm model}(x)]^2}{\sum_i^{N_{\rm fit}} [\chi_i^{\rm measured}]^2}$$

reduced chi-square: scale by the "degrees of freedom" :

$$\chi_\nu^2 = \chi^2/(N_{\rm idp} - N_{\rm varys})$$

Akaike Information Criterion: Also weights to account for degrees of freedom in fit:

$$\mathrm{AIC} = N_{\rm data} \log(\chi^2/N_{\rm data}) + 2N_{\rm varys}$$

For a "Good Fit", χ_ν^2 should be ~ 1.
This assumes that we have an accurate estimate of ϵ which never really happens!

Other Fitting Statistics

Other "goodness-of-fit statistics":

chi-square: As before:

$$\chi^2 = \frac{N_{\mathrm{idp}}}{\epsilon^2 N_{\mathrm{fit}}} \sum_i^{N_{\mathrm{fit}}} [\chi_i^{\mathrm{measured}} - \chi_i^{\mathrm{model}}(x)]^2$$

R-factor: $\mathcal{R}$ gives a "fractional misfit" (and not scaled by the data uncertainty ϵ):

$$\mathcal{R} = \frac{\sum_i^{N_{\mathrm{fit}}} [\chi_i^{\mathrm{measured}} - \chi_i^{\mathrm{model}}(x)]^2}{\sum_i^{N_{\mathrm{fit}}} [\chi_i^{\mathrm{measured}}]^2}$$

reduced chi-square: scale by the "degrees of freedom" :

$$\chi_\nu^2 = \chi^2/(N_{\mathrm{idp}} - N_{\mathrm{varys}})$$

Akaike Information Criterion: Also weights to account for degrees of freedom in fit:

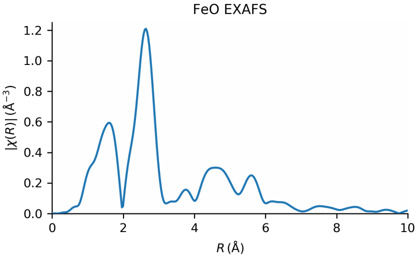
$$\mathrm{AIC} = N_{\mathrm{data}} \log(\chi^2/N_{\mathrm{data}}) + 2N_{\mathrm{varys}}$$

For a "Good Fit", χ_ν^2 should be ~ 1.
This assumes that we have an accurate estimate of ϵ which never really happens!

Propagation of uncertainties in $\chi(k)$

Estimating uncertainties in $\chi(k)$ has always been a challenge.

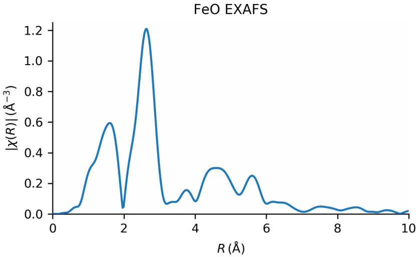
We have (by default) estimated the uncertainty in $\chi(k)$ as *white noise* (Newville, Boyanov, and Sayers, *J Synch Rad*, 1999), using $\chi(R)$ between [15, 25] Å.

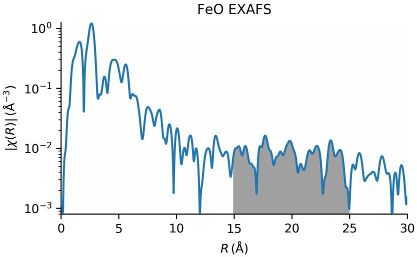


Propagation of uncertainties in $\chi(k)$

Estimating uncertainties in $\chi(k)$ has always been a challenge.

We have (by default) estimated the uncertainty in $\chi(k)$ as *white noise* (Newville, Boyanov, and Sayers, *J Synch Rad*, 1999), using $\chi(R)$ between [15, 25] Å.





The "high-R" portion of $\chi(R)$ can estimate the "white noise" in the data pretty well.

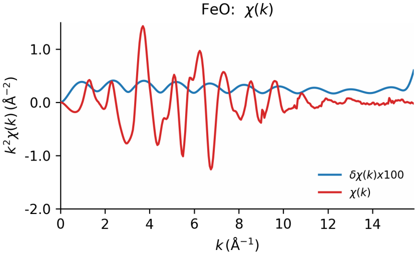
This is easy to do, but we know it misses an important component:

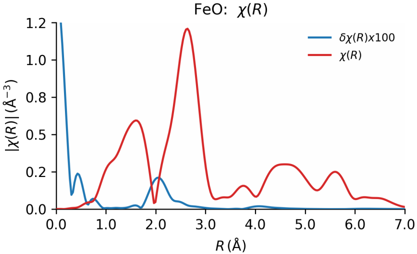
uncertainties from background subtraction

Uncertainties in $\chi(k)$ from background subtraction

We can propagate the uncertainties from the fit of the background spline to estimate the uncertainty in $\chi(k)$ from the background subtraction.

This is *not white noise*. In fact, it tends to have a peak somewhat above $2R_{\rm bkg}$

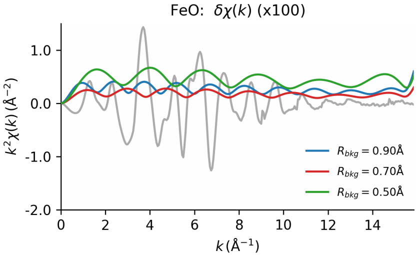


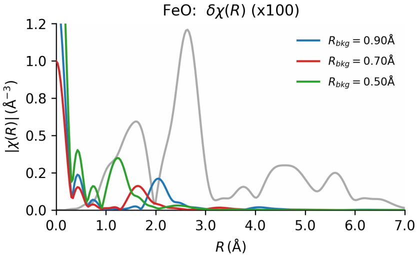


Uncertainties in $\chi(k)$ from background subtraction

We can propagate the uncertainties from the fit of the background spline to estimate the uncertainty in $\chi(k)$ from the background subtraction.

This is *not white noise*. In fact, it tends to have a peak somewhat above $2R_{\rm bkg}$

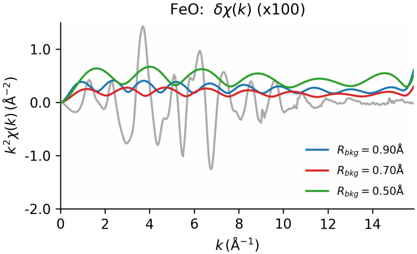

FeO: $\delta\chi(k)$ (x100)

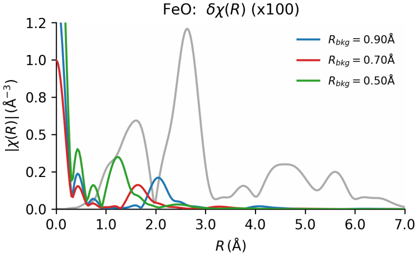

FeO: $\delta\chi(R)$ (x100)

Uncertainties in $\chi(k)$ from background subtraction

We can propagate the uncertainties from the fit of the background spline to estimate the uncertainty in $\chi(k)$ from the background subtraction.

This is *not white noise*. In fact, it tends to have a peak somewhat above $2R_{\rm bkg}$


FeO: $\delta\chi(k)$ (x100)


FeO: $\delta\chi(R)$ (x100)

Using this $\delta\chi(k)$ array as the estimate of the uncertainty of the EXAFS $\chi(k)$ reduces the χ^2 statistic by $2\times$ or more.

This is now the default approach in LARCH.

Error Bars: the uncertainties in the fit variables

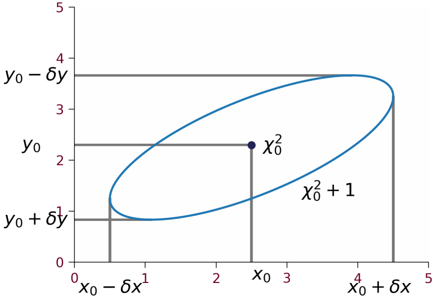
A fit finds the "best-fit" set of values for the variables $\{x, y, \ldots\}$: these give the lowest $\chi^2 = \chi_0^2$.

Uncertainties in Parameters x are estimated by increasing the χ^2 by 1:

Error Bars: the uncertainties in the fit variables

A fit finds the "best-fit" set of values for the variables $\{x, y, \ldots\}$: these give the lowest $\chi^2 = \chi^2_0$.

Uncertainties in Parameters x are estimated by increasing the χ^2 by 1:



Some Parameters are *Correlated*:

Changing the value for parameter x away from its best value will change the best value for another parameter, y.

For EXAFS, (R, E_0) and (N, σ^2) are usually very highly correlated (> 0.85).

Increasing χ^2 by 1 assumes we have a "Good Fit", with $\chi^2_\nu \approx 1$.

We typically have $\chi^2_\nu \sim 10$, so we increase the best χ^2 by χ^2_ν to estimate error bars.

The reported uncertainties do take the correlation into account!

More rigorous methods for uncertainty analysis is available from the LARCH Python code.

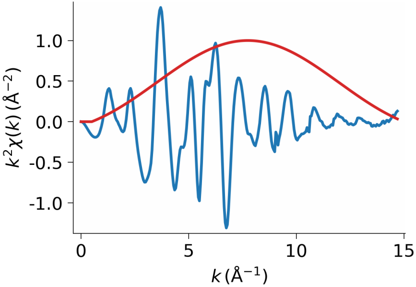
Fitting in R- or k-space: What do we model?

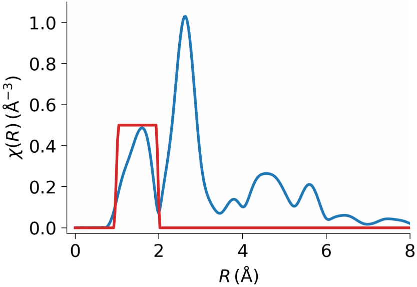
The χ^2 definition didn't say anything about what our data $\chi_i^{\mathrm{measured}}$ actually is . . .

Fitting in R- or k-space: What do we model?

The χ^2 definition didn't say anything about what our data $\chi_i^{\rm measured}$ actually is . . .

We usually fit in R-space, so that we can select which "shells" to ignore:

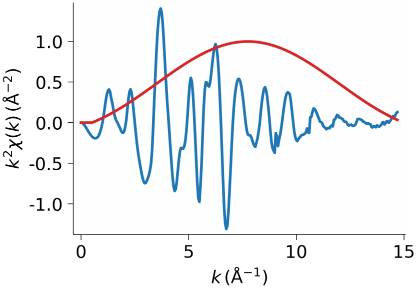


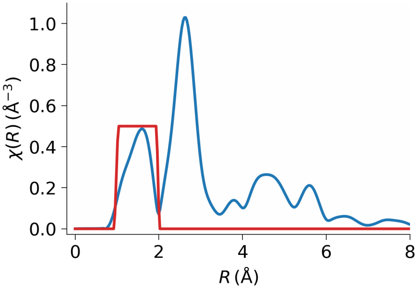


Fitting in R- or k-space: What do we model?

The χ^2 definition didn't say anything about what our data $\chi_i^{\rm measured}$ actually is . . .

We usually fit in R-space, so that we can select which "shells" to ignore:





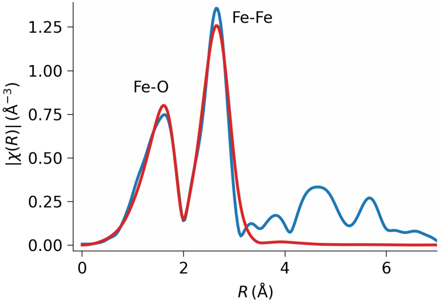
Fitting $\chi(R)$ (both real and imaginary parts!) gives more meaningful fit statistics – we know that we're not fitting all the spectral features.

Plus: We can have $\chi_i^{\rm measured}$ extend over multiple data sets, multiple k-weightings, etc.

as long as we generate the corresponding $\chi_i^{\rm model}(x)$ to match these data.

EXAFS Analysis: Second Shell of FeO

Adding the 2nd shell Fe – *feffNNNN.dat* for Fe-Fe – and refining R, N, σ^2:

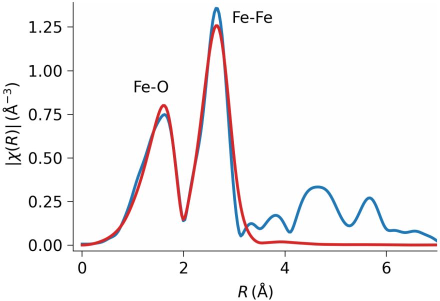


$|\chi(R)|$ data for FeO (blue), and fit of $1^{\rm st}$ and $2^{\rm nd}$ shells (red).

These results are consistent with the known values for FeO:
6 O at 2.14Å, 12 Fe at 3.03Å.

EXAFS Analysis: Second Shell of FeO

Adding the 2nd shell Fe – `feffNNNN.dat` for Fe-Fe – and refining R, N, σ^2:



$|\chi(R)|$ data for FeO (blue), and fit of $1^{\rm st}$ and $2^{\rm nd}$ shells (red).

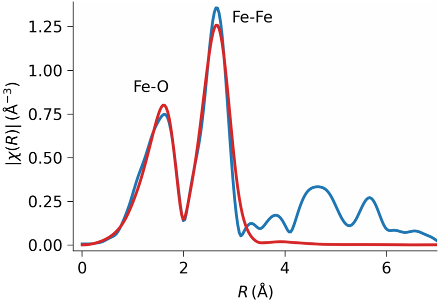
These results are consistent with the known values for FeO:
6 O at 2.14Å, 12 Fe at 3.03Å.

Fit results: Statistics: $R \approx 0.01$ $\chi^2_\nu \approx 3.$

Shell	N	R (Å)	σ^2 (Å^2)	ΔE_0 (eV)
Fe-O	4.6(0.6)	2.11(.01)	0.011(.002)	1.8(0.7)
Fe-Fe	14.1(1.7)	3.08(.01)	0.015(.002)	1.8(0.7)

EXAFS Analysis: Second Shell of FeO

Adding the 2nd shell Fe – *feffNNNN.dat* for Fe-Fe – and refining R, N, σ^2:



$|\chi(R)|$ data for FeO (blue), and fit of 1^{st} and 2^{nd} shells (red).

These results are consistent with the known values for FeO:
6 O at 2.14Å, 12 Fe at 3.03Å.

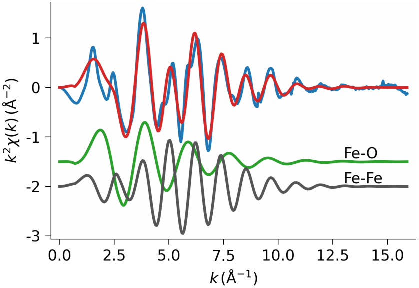
Fit results: Statistics: $R \approx 0.01$ $\chi^2_\nu \approx 3$.

Shell	N	R (Å)	σ^2 (Å^2)	ΔE_0 (eV)
Fe-O	4.6(0.6)	2.11(.01)	0.011(.002)	1.8(0.7)
Fe-Fe	14.1(1.7)	3.08(.01)	0.015(.002)	1.8(0.7)

These are typical even for a “very good fit” on known structures.
The calculation for $f(k)$ and $\delta(k)$ are good, but not perfect!

EXAFS Analysis: Second Shell of FeO

Other views of the data and fit:

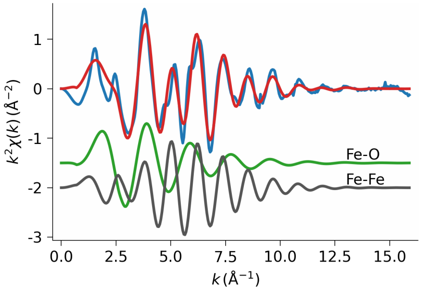


The Fe-Fe EXAFS extends to higher-k than the Fe-O EXAFS.

Even in this simple system, there is some *overlap* of shells in R-space.

EXAFS Analysis: Second Shell of FeO

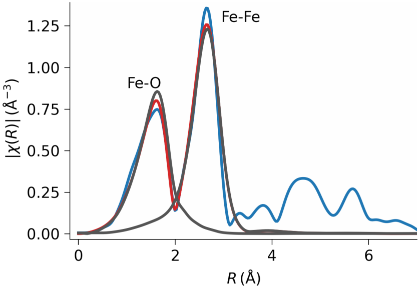
Other views of the data and fit:

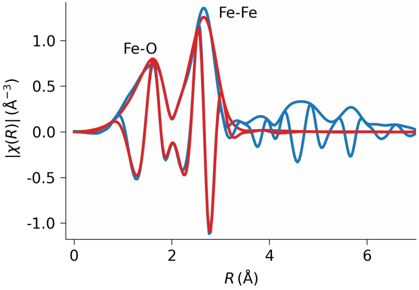


The Fe-Fe EXAFS extends to higher-k than the Fe-O EXAFS.

Even in this simple system, there is some *overlap* of shells in R-space.

The fit in $\mathrm{Re}[\chi(\mathrm{R})]$ look especially good – this is how the fits are done.





Path Parameters: what can we vary in a fit?

The EXAFS Equation has at least 4 adjustable parameters *Per Path*:

$$E_0,\ NS_0^2,\ R, \text{ and } \sigma^2.$$

But $N_{\rm idp}$ is low:

$$N_{\rm idp} = 8 \ \text{ for } \Delta R = 1\,\text{Å} \ \text{ and } \ \Delta k = 12.5\,\text{Å}^{-1}$$

Path Parameters: what can we vary in a fit?

The EXAFS Equation has at least 4 adjustable parameters *Per Path*:

$$E_0,\ NS_0^2,\ R, \text{ and } \sigma^2.$$

But $N_{\rm idp}$ is low:

$$N_{\rm idp} = 8 \ \ \text{for}\ \Delta R = 1\,\text{Å} \ \ \text{and} \ \ \Delta k = 12.5\,\text{Å}^{-1}$$

For simple crystalline structures with well-isolated, single-scattering:
path (like FeO), it's OK to fit N, R, σ^2, and E_0 for every path.

For more complicated problems, we need a way to limit the number of parameters varied.

We might *want* to impose relationships between parameters to get more meaningful results. . .

Constraints and Generalized Variables

Instead of varying the Path Parameters directly, we write them in terms of *Generalized Variables*. This allows simple *Constraints* and model building:

Parameter=Variable

```
# one variable e0 for 2 paths
params = group(e0 = guess(1.0), ...)

path1  = feffpath('feo.dat', e0='e0')
path2  = feffpath('fefe.dat', e0='e0')
```

Constraints and Generalized Variables

Instead of varying the Path Parameters directly, we write them in terms of *Generalized Variables*. This allows simple *Constraints* and model building:

Parameter=Variable

```
# one variable e0 for 2 paths
params = group(e0 = guess(1.0), ...)

path1  = feffpath(’feo.dat’, e0=’e0’)
path2  = feffpath(’fefe.dat’, e0=’e0’)
```

Mixed Coordination Shell

```
# mix O and S in 1st coordination shell
params = group(s02   = param(0.80, vary=False),
               sfrac = guess(0.5))

path1 = feffpath(’feo.dat’, s02=’s02*sfrac’)
path2 = feffpath(’fes.dat’, s02=’s02*(1-sfrac)’)
```

Constraints and Generalized Variables

Instead of varying the Path Parameters directly, we write them in terms of *Generalized Variables*. This allows simple *Constraints* and model building:

Parameter=Variable

```
# one variable e0 for 2 paths
params = group(e0 = guess(1.0), ...)

path1  = feffpath(’feo.dat’, e0=’e0’)
path2  = feffpath(’fefe.dat’, e0=’e0’)
```

Mixed Coordination Shell

```
# mix O and S in 1st coordination shell
params = group(s02   = param(0.80, vary=False),
               sfrac = guess(0.5))

path1 = feffpath(’feo.dat’, s02=’s02*sfrac’)
path2 = feffpath(’fes.dat’, s02=’s02*(1-sfrac)’)
```

Einstein Temperature

```
# Use 1 ‘‘theta’’ to set sigma2 for multiple paths

params = group(amp=param(1, vary=True),
               theta=param(250, min=0, vary=True), ...)

path1_100K = feffpath(’fefe.dat’, s02=’amp’, ...,
                      sigma2=’sigma2_eins(100, theta)’)

path1_200K = feffpath(’fefe.dat’, s02=’amp’, ...,
                      sigma2=’sigma2_eins(200, theta)’)

path1_300K = feffpath(’fefe.dat’, s02=’amp’, ...,
                      sigma2=’sigma2_eins(300, theta)’)
```

This allows us to use *Prior Knowledge* into the data analysis, and consider more complicated problems:

- force one R for the same bond for data taken from different edges.
- model complex distortions (height of a sorbed atom above a surface).

Also: Each Parameter can have upper and lower bounds, say to ensure that $\sigma^2 > 0$.

Example: Cu metal at 3 temperature

A very simple example of a Multi-Data-Set Fit:
Cu metal, at 3 different temperatures: 10K, 50K 150K.

Path Parameters:

- E_0: Same for all T
- S_0^2 Same for all T
- R: expands linearly with T (slope + offset).
- σ^2: goes as Einstein temperature (as before).

12 parameters become 5.

Fit range:

$$R = [1.60, 2.75]\,\text{Å}$$

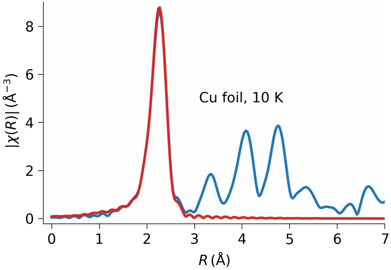
$$k = [1.50, 18.50]\,\text{Å}^{-1}$$

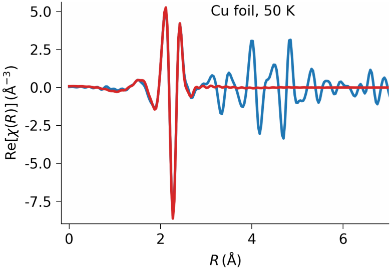
Cu at three temperatures

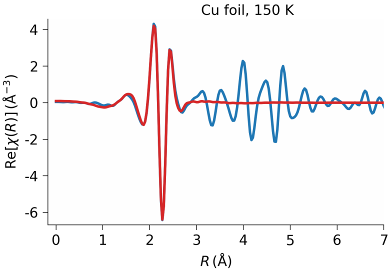
```
# define fitting parameter group
pars = group(amp       = param(1, vary=True),
             del_e0    = guess(2.0),
             theta     = param(250, min=10, vary=True),
             dr_off    = guess(0),
             dr_slope = guess(0) )

# define 3 Feff Path, give expressions for Path Parameters
path1_10  = feffpath(’feff0001.dat’,
                     s02=’amp’, e0=’del_e0’,
                     deltar=’dr_off + 10*dr_slope’,
                     sigma2=’sigma2_eins(10, theta)’)

path1_50  = feffpath(’feff0001.dat’,
                     s02=’amp’, e0=’del_e0’,
                     deltar=’dr_off + 50*dr_slope’,
                     sigma2=’sigma2_eins(50, theta)’)

path1_150 = feffpath(’feff0001.dat’,
                     s02=’amp’, e0=’del_e0’,
                     deltar=’dr_off + 150*dr_slope’,
                     sigma2=’sigma2_eins(150, theta)’)
```

Example: Cu metal Results

`amp`	$0.91(0.08)$
`theta`	$233.5(19.6)\,\mathrm{K}$
`del_e0`	$0.4(1.3)\,\mathrm{eV}$
`dr_off`	$0.002(0.003)\,\mathrm{Å/K}$
`dr_slope`	$0.5(1.8) \times 10^{-5}\,\mathrm{Å}$







Structural Disorder and the Pair Distribution Function

An EXAFS measurement averages billions of *snapshots* of the local structure:

- Each absorbed x-ray generates 1 photo-electron.
- the photo-electron / core-hole pair lives for about 10^{-15} s – much faster than the thermal vibrations (10^{-12} s).
- An EXAFS measurement samples 10^4 (dilute fluorescence) to 10^{10} absorbed x-rays for each energy point.

So far, we've put this in the EXAFS Equation as $\chi \sim N \exp(-2k^2\sigma^2)$

Structural Disorder and the Pair Distribution Function

An EXAFS measurement averages billions of *snapshots* of the local structure:

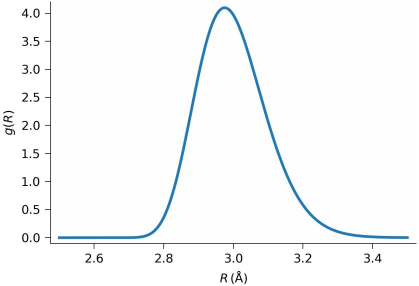
- Each absorbed x-ray generates 1 photo-electron.
- the photo-electron / core-hole pair lives for about 10^{-15} s – much faster than the thermal vibrations (10^{-12} s).
- An EXAFS measurement samples 10^{4} (dilute fluorescence) to 10^{10} absorbed x-rays for each energy point.

So far, we've put this in the EXAFS Equation as $\chi \sim N \exp(-2k^2\sigma^2)$

More generally, EXAFS samples the

Partial Pair Distribution Function

$g(R)$ = probability that an atom is a distance R away from the absorber.



EXAFS and The Pair Distribution Function

To fully account for a highly disordered local structure, we should use

$$\chi(k) = \left\langle \sum_j \frac{f_j(k) e^{i2kR_j + \delta_j(k)}}{kR_j^2} \right\rangle$$

where $\langle x \rangle = \int dR\, x\, g(R) / \int dR\, g(R)$ – averaging over the billions+ of snapshots.

R won't change too much, so we'll neglect the changes to $1/R^2$:

$$\chi \approx \sum_j f_j(k) \frac{e^{i\delta_j(k)}}{kR_j^2} \left\langle e^{i2kR_j} \right\rangle$$

each path in the sum now has a $g(R)$ with respect to the absorbing atom.

EXAFS and The Pair Distribution Function

To fully account for a highly disordered local structure, we should use

$$\chi(k) = \left\langle \sum_j \frac{f_j(k) e^{i2kR_j + \delta_j(k)}}{kR_j^2} \right\rangle$$

where $\langle x \rangle = \int dR \, x \, g(R) / \int dR \, g(R)$ – averaging over the billions$+$ of snapshots.

R won't change too much, so we'll neglect the changes to $1/R^2$:

$$\chi \approx \sum_j f_j(k) \frac{e^{i\delta_j(k)}}{kR_j^2} \left\langle e^{i2kR_j} \right\rangle$$

each path in the sum now has a $g(R)$ with respect to the absorbing atom.

EXAFS and The Pair Distribution Function

To fully account for a highly disordered local structure, we should use

$$\chi(k) = \left\langle \sum_j \frac{f_j(k) e^{i2kR_j + \delta_j(k)}}{kR_j^2} \right\rangle$$

where $\langle x \rangle = \int dR \, x \, g(R) / \int dR \, g(R)$ – averaging over the billions+ of snapshots.

R won't change too much, so we'll neglect the changes to $1/R^2$:

$$\chi \approx \sum_j f_j(k) \frac{e^{i\delta_j(k)}}{kR_j^2} \left\langle e^{i2kR_j} \right\rangle$$

each path in the sum now has a $g(R)$ with respect to the absorbing atom.

The *the cumulant expansion* relates $\langle e^x \rangle$ to $\langle x \rangle$, the moments of $g(x)$:

$$\left\langle e^{i2kR} \right\rangle = \exp \left[\sum_{n=1}^{\infty} \frac{(2ik)^n}{n!} C_n \right].$$

The Cumulants and Moments of a Distribution Function

The cumulants C_n of $g(R)$ are related to the moments of $g(R)$: $\langle r^n \rangle$, with $r = R - R_0$ and R_0 is the centroid of the distribution:

$$\begin{array}{lll}
C_1 = \Delta R & \texttt{deltar} & = \langle r \rangle \\
C_2 = \sigma^2 & \texttt{sigma2} & = \langle r^2 \rangle - \langle r \rangle^2 \\
C_3 & \texttt{third} & = \langle r^3 \rangle - 3\langle r^2 \rangle \langle r \rangle + 2\langle r \rangle^3 \\
C_4 & \texttt{fourth} & = \langle r^4 \rangle - 3\langle r^2 \rangle^2 - 4\langle r^3 \rangle \langle r \rangle + 12\langle r^2 \rangle \langle r \rangle^2 - 6\langle r \rangle^4
\end{array}$$

C_3 (the *third cumulant*) can be important in many cases.

The Cumulants and Moments of a Distribution Function

The cumulants C_n of $g(R)$ are related to the moments of $g(R)$: $\langle r^n \rangle$, with $r = R - R_0$ and R_0 is the centroid of the distribution:

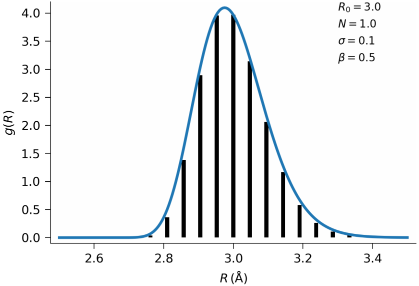
$$\begin{array}{lll}
C_1 = \Delta R & \texttt{deltar} & = \langle r \rangle \\
C_2 = \sigma^2 & \texttt{sigma2} & = \langle r^2 \rangle - \langle r \rangle^2 \\
C_3 & \texttt{third} & = \langle r^3 \rangle - 3\langle r^2 \rangle \langle r \rangle + 2\langle r \rangle^3 \\
C_4 & \texttt{fourth} & = \langle r^4 \rangle - 3\langle r^2 \rangle^2 - 4\langle r^3 \rangle \langle r \rangle + 12\langle r^2 \rangle \langle r \rangle^2 - 6\langle r \rangle^4
\end{array}$$

C_3 (the *third cumulant*) can be important in many cases.

But: Sometimes, the cumulant expansion isn't good enough. One can also build models by using paths spaced in R (say, at 0.2 Å steps), and model the amplitude of each Path with a distribution like (following GNXAS):

$$g(R, N, R_0, \sigma, \beta) = \frac{2N[e^{-\alpha}\alpha^{q-1}]}{\sigma\beta\Gamma(q)}$$

where $\alpha = q + 2(R - R_0)/(\beta\sigma)$, and $q = 4/\beta^2$



EXAFS Data Analysis with FEFF

Using FEFF to model EXAFS mostly means paying attention to:

- N_{idp} – not very many Parameters can be varied for a limited k and R range.
- Always look at the uncertainties in the Parameters, not just best-fit values.
- Check (or require in the fit) that $\sigma^2 > 0$, $N > 0$.
- Think about how you might combine Parameters for different Paths, ideally making a physical model.
- Try a third cumulant now and then – it might be needed.
- For very disordered systems, cumulants might not be enough.

More information on X-rays and X-ray Absorption Spectroscopy:

https://xafs.xrayabsorption.org/

Fundamentals of XAFS M. Newville, Reviews in Mineralogy & Geochemistry **78**, 2014.

Introduction to XAFS G. Bunker, Cambridge Univ Press, 2010.

XAFS for Everyone S. Calvin, CRC Press, 2013.

Elements of Modern X-ray Physics J. Als-Nielsen & D. McMorrow, John Wiley & Sons. 2001